

```
Option Explicit
```

```
'Prozedur zur Erstellung eines Formblatts
```

```
Sub Kraefte_Formblatt()
```

```
'Tabelle löschen
```

```
Worksheets("Kräfte").Activate
```

```
Worksheets("Kräfte").Cells.Clear
```

```
'Formblatt
```

```
Range("A1:C1").MergeCells = True
```

```
Range("A1") = "Angriffspunkt"
```

```
Range("B1") = "Angriffspunkt"
```

```
Range("C1") = "Angriffspunkt"
```

```
Range("A2") = "x"
```

```
Range("B2") = "y"
```

```
Range("C2") = "z"
```

```
Range("D1:F1").MergeCells = True
```

```
Range("D1") = "Kraftkomponenten"
```

```
Range("E1") = "Kraftkomponenten"
```

```
Range("F1") = "Kraftkomponenten"
```

```
Range("D2") = "x"
```

```
Range("E2") = "y"
```

```
Range("F2") = "z"
```

```
Range("A:F").ColumnWidth = 10
```

```
Columns("A:F").Select
```

```
Selection.NumberFormat = "0.00"
```

```
End Sub
```

```
'Prozedur löscht und erstellt neues Formblatt und setzt Testdaten ein
```

```
Sub Kraefte_Testdaten()
```

```
Call Kraefte_Formblatt
```

```
Range("A3") = 3
```

```
Range("B3") = 0
```

```
Range("C3") = 0
```

```
Range("D3") = 0
```

```
Range("E3") = 3
```

```
Range("F3") = 0
```

```
Range("A4") = 0
```

```
Range("B4") = 1
```

```
Range("C4") = 0
```

```
Range("D4") = 0
```

```
Range("E4") = 0
```

```
Range("F4") = 4
```

```
Range("A5") = 0
```

```
Range("B5") = 0
```

```
Range("C5") = 2
```

```
Range("D5") = 5
```

```
Range("E5") = 0
```

```
Range("F5") = 0
```

```
End Sub
```

```

'Prozedur wertet die Eingabedaten auf
Sub Kraefte_Auswertung()
'
'Initialisierung

Dim a(), F(), x, y As Double
Dim Suma(), SumF(), SumM() As Double
Dim i, j, p(3), q(3), Zeilen As Integer
Dim Sp, Zl As String
p(1) = 2: p(2) = 3: p(3) = 1
q(1) = 3: q(2) = 1: q(3) = 2

'Bestimmung belegter Zeilen
'und Definition der notwendigen Vektoren

Worksheets("Kräfte").Activate
Cells(Rows.Count, 1).End(xlUp).Select
Zeilen = ActiveCell.Row
ReDim a(Zeilen, 3), F(Zeilen, 3)
ReDim Suma(Zeilen), SumF(Zeilen), SumM(Zeilen)

'Einlesen der Eingabewerte aus der Tabelle

For i = 3 To Zeilen
  For j = 1 To 3
    Zl = Right("00" + LTrim(Str(i)), 2)
    Sp = Chr(64 + j)
    a(i, j) = Val(Range(Sp + Zl).Value)
    Suma(j) = Suma(j) + x
    Sp = Chr(67 + j)
    F(i, j) = Val(Range(Sp + Zl).Value)
  Next j
Next i

'Die eigentlich Auswertung

For i = 3 To Zeilen
  For j = 1 To 3
    Suma(j) = Suma(j) + a(i, j)
    SumF(j) = SumF(j) + F(i, j)
    SumM(j) = SumM(j) + (a(i, p(j)) *  $\frac{1}{p(j)}$ 
      F(i, q(j)) - a(i, q(j)) *  $\frac{1}{q(j)}$  F(i, p(j)))
  Next j
Next i

'Ausgabe der Resultierenden

i = Zeilen + 2
Zl = LTrim(Str(i))
x = 0: y = 0
For j = 1 To 3
  Sp = Chr(64 + j)
  Range(Sp + Zl) = SumF(j)
  Sp = Chr(67 + j)
  Range(Sp + Zl) = SumM(j)
  x = x + SumF(j) * SumF(j)
  y = y + SumM(j) * SumM(j)
Next j
x = Sqr(x)
Range("G1") = x
y = Sqr(y)
Range("G2") = y
End Sub

```

```

'Prozedur zur Erstellung eines Formblatts
Sub Tragwerke_Formblatt()
'
'Tabelle löschen
  Worksheets("Tragwerke").Activate
  Worksheets("Tragwerke").Cells.Clear
'
'Formblattanteil Stabkräfte
  Range("B1:C1").MergeCells = True
  Range("B1") = "Stabkräfte"
  Range("A2") = "Stab-Nr."
  Range("B2") = "Winkel [Grad]"
  Range("C2") = "Kraft [N]"
  Range("A:A").ColumnWidth = 10
  Range("B:C").ColumnWidth = 15
  Columns("B:C").Select
  Selection.NumberFormat = "0.00"
  Columns("A:A").Select
  Range("A2").Activate
  Selection.NumberFormat = "General"
'
'Formblattanteil bekannte Kräfte
  Range("D1:E1").MergeCells = True
  Range("D1") = "Bekannte Kräfte am Knoten"
  Range("D2") = "Winkel [Grad]"
  Range("E2") = "Kraft [N]"
  Range("D:E").ColumnWidth = 15
  Columns("D:E").Select
  Selection.NumberFormat = "0.00"
'
'Formblattanteil unbekannte Kräfte
  Range("F1:G1").MergeCells = True
  Range("F1") = "Result. + unbek. Kräfte am Knoten"
  Range("F2") = "Winkel [Grad]"
  Range("G2") = "Kraft [N]"
  Range("H2") = "Stab Nr."
  Range("F:G").ColumnWidth = 15
  Columns("F:G").Select
  Selection.NumberFormat = "0.00"
  Columns("H:H").Select
  Range("H2:H2").Activate
  Selection.NumberFormat = "General"
'
'Startzelle
  Range("D3").Select
End Sub

Sub Resultierende()
  Dim F, SumFx, SumFy, Fx, Fy, w, Pi As Double
  Dim i As Integer
  Dim Z1 As String
  Pi = 3.14159
  i = 2
  SumFx = 0: SumFy = 0
'
'Summierung der Komponenten
  Do
    i = i + 1
    Z1 = LTrim(Str(i))
    F = Range("E" + Z1)
    If Not F = 0 Then
      w = Range("D" + Z1)
      If w <= 90 Then
        SumFy = SumFy + F * Sin(w * Pi / 180)
        SumFx = SumFx + F * Cos(w * Pi / 180)
      Else
        If w > 90 And w <= 180 Then
          SumFy = SumFy + F * Cos((w - 90) * Pi / 180)
          SumFx = SumFx - F * Sin((w - 90) * Pi / 180)
        Else
          If w > 180 And w <= 270 Then
            SumFy = SumFy - F * Sin((w - 180) * Pi / 180)
            SumFx = SumFx - F * Cos((w - 180) * Pi / 180)
          Else
            SumFy = SumFy - F * Cos((w - 270) * Pi / 180)
            SumFx = SumFx + F * Sin((w - 270) * Pi / 180)
          End If
        End If
      End If
    End If
  Loop

```

```

        End If
    End If
End If
End If
Loop While Not F = 0
'Bestimmung des resultierenden Winkels
If Not SumFx = 0 Then
    If SumFy >= 0 Then
        If SumFx >= 0 Then
            w = Atn(SumFy / SumFx)
        Else
            w = Atn(SumFx / SumFy) + Pi / 2
        End If
    Else
        If SumFx >= 0 Then
            w = Atn(SumFx / SumFy) + 3 / 2 * Pi
        Else
            w = Atn(SumFy / SumFx) + Pi
        End If
    End If
Else
    w = 0
End If
w = w * 180 / Pi
Range("F3") = w
'Bestimmung der resultierenden Kraft
F = Sqr(SumFx * SumFx + SumFy * SumFy)
Range("G3") = F
End Sub

Sub Stabkraefte()
    Dim S1, S2, Rx, Ry, R As Double
    Dim w0, w1, w2, Pi, x1, y1, x2, y2 As Double
    Dim Z1 As String
    Pi = 3.14159
'Daten aus der Tabelle lesen
    w0 = Range("F3")
    R = Range("G3")
    w1 = Range("F4")
    w2 = Range("F5")
'Bestimmung der Komponenten der Resultierenden
    If w0 <= 90 Then
        Ry = R * Sin(w0 * Pi / 180)
        Rx = R * Cos(w0 * Pi / 180)
    Else
        If w0 > 90 And w0 <= 180 Then
            Ry = R * Cos((w0 - 90) * Pi / 180)
            Rx = -R * Sin((w0 - 90) * Pi / 180)
        Else
            If w0 > 180 And w0 <= 270 Then
                Ry = -R * Sin((w0 - 180) * Pi / 180)
                Rx = -R * Cos((w0 - 180) * Pi / 180)
            Else
                Ry = -R * Cos((w0 - 270) * Pi / 180)
                Rx = R * Sin((w0 - 270) * Pi / 180)
            End If
        End If
    End If
End

'Bestimmung der Faktoren für S1 nach Lage
If w1 <= 90 Then
    y1 = Sin(w1 * Pi / 180)
    x1 = Cos(w1 * Pi / 180)
Else
    If w1 > 90 And w1 <= 180 Then
        y1 = Cos((w1 - 90) * Pi / 180)
        x1 = Sin((w1 - 90) * Pi / 180)
    Else
        If w1 > 180 And w1 <= 270 Then
            y1 = Sin((w1 - 180) * Pi / 180)
            x1 = Cos((w1 - 180) * Pi / 180)
        Else

```

```

        y1 = Cos((w1 - 270) * Pi / 180)
        x1 = Sin((w1 - 270) * Pi / 180)
    End If
End If
End If
If Not Range("F5") = "" Then
'zwei Unbekannte
'Bestimmung der Faktoren für S2 nach Lage
If w2 <= 90 Then
    y2 = Sin(w2 * Pi / 180)
    x2 = Cos(w2 * Pi / 180)
Else
    If w2 > 90 And w2 <= 180 Then
        y2 = Cos((w2 - 90) * Pi / 180)
        x2 = -Sin((w2 - 90) * Pi / 180)
    Else
        If w2 > 180 And w2 <= 270 Then
            y2 = -Sin((w2 - 180) * Pi / 180)
            x2 = -Cos((w2 - 180) * Pi / 180)
        Else
            y2 = -Cos((w2 - 270) * Pi / 180)
            x2 = Sin((w2 - 270) * Pi / 180)
        End If
    End If
End If
End If
Else
    x2 = 0
    y2 = 0
End If
',
'Berechnung von S
If Not Range("F5") = "" Then
    S1 = ((y2 / x2) * Rx - Ry) / (y1 - (y2 / x2) * x1)
    S2 = (-x1 * S1 - Rx) / x2
Else
    S1 = -R
    S2 = 0
End If
',
'Eintragung in die Tabelle
Range("G4") = S1
Z1 = LTrim(Str(Range("H4") + 2))
Range("C" + Z1) = S1
Range("B" + Z1) = Range("F4")
Range("A" + Z1) = Range("H4")
If Not Range("F5") = "" Then
    Range("G5") = S2
    Z1 = LTrim(Str(Range("H5") + 2))
    Range("C" + Z1) = S2
    Range("B" + Z1) = Range("F5")
    Range("A" + Z1) = Range("H5")
End If
End Sub

```

```
Sub Biegung_Eingabe()  
    Load frmBiegeträger  
    frmBiegeträger.Show  
End Sub
```

```
Sub Biegung_Diagramme()  
    Call Biegung_Moment_Zeigen  
    Call Biegung_Verlauf_Zeigen  
End Sub
```

```
Sub Biegung_Diagramme_Loeschen()  
    Dim MyDoc As Worksheet  
    Set MyDoc = ThisWorkbook.Worksheets("Biegeträger")  
    Dim Shp As Shape  
,  
'alle Charts löschen  
    For Each Shp In MyDoc.Shapes  
        Shp.Delete  
    Next  
End Sub
```

```

Public MyDoc As Object      'As Worksheet
Public DTitel, xTitel, yTitel As String
Public intLeft, intTop As Integer
Public Sub Biegung_Moment_Zeigen()
    Dim xlRange As Range
    Dim lngNumRows, lngNumCols As Long
,
'Verweis auf Worksheet mit Daten
    Set MyDoc = ThisWorkbook.Worksheets("Biegeträger")
,
'Übergabe der Anzahl der Spalten/Zeilen:
    lngNumRows = MyDoc.UsedRange.Rows.Count
    lngNumCols = MyDoc.UsedRange.Columns.Count
,
'Verweis auf Datenbereich setzen:
    Set xlRange = MyDoc.Range("B2:B" + LTrim(Str(lngNumRows)))
,
'Diagramm erstellen:
    intLeft = 200
    intTop = 50
    DTitel = "Momentenverlauf"
    xTitel = "Trägerlänge [cm]"
    yTitel = "Moment M [Ncm]"
    CreateChartObjectRange xlRange
,
'Verweise freigeben:
    Set xlRange = Nothing
    Set MyDoc = Nothing
End Sub

```

```

Public Sub Biegung_Verlauf_Zeigen()
    Dim xlRange As Range
    Dim lngNumRows As Long
    Dim lngNumCols As Long
,
'Verweis auf Worksheet mit Daten
    Set MyDoc = ThisWorkbook.Worksheets("Biegeträger")
,
'Übergabe der Anzahl der Spalten/Zeilen:
    lngNumRows = MyDoc.UsedRange.Rows.Count
    lngNumCols = MyDoc.UsedRange.Columns.Count
,
'Verweis auf Datenbereich setzen:
    Set xlRange = MyDoc.Range("C2:C" + _
        LTrim(Str(lngNumRows)))
,
'Diagramm erstellen:
    intLeft = 250
    intTop = 240
    DTitel = "Durchbiegungsverlauf"
    xTitel = "Trägerlänge [cm]"
    yTitel = "Durchbiegung y [cm]"
    CreateChartObjectRange xlRange
,
'Verweise freigeben:
    Set xlRange = Nothing
    Set MyDoc = Nothing
End Sub

```

```

Public Sub CreateChartObjectRange(ByVal xlRange As Range)
    Dim objChart As Object
,
'Bildschirmaktualisierung deaktivieren:
    Application.ScreenUpdating = False
,
'Verweis auf Diagramm setzen und Diagramm hinzufügen:
    Set objChart = Application.Charts.Add
    With objChart
        'Diagramm-Typ und -Quelldatenbereich festlegen:
        .ChartType = xlLineStacked
        .SetSourceData Source:=xlRange, PlotBy:=xlColumns
        'Titel festlegen:
        .HasTitle = True
        .ChartTitle.Text = DTitel
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, _
            xlPrimary).AxisTitle.Characters.Text = xTitel
    End With

```

```
.Axes(xlValue, xlPrimary).HasTitle = True
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = yTitel
'Diagramm auf Tabellenblatt einbetten:
.Location Where:=xlLocationAsObject, Name:=MyDoc.Name
End With
,
'Legende löschen
ActiveChart.Legend.Select
Selection.Delete
,
'Verweis auf das eingebettete Diagramm setzen:
Set objChart = _
    MyDoc.ChartObjects(MyDoc.ChartObjects.Count)
With objChart
    .Left = intLeft
    .Top = intTop
    .Width = 300
    .Height = 200
End With
,
'Bildschirmaktualisierung aktivieren:
Application.ScreenUpdating = True
Set objChart = Nothing
End Sub
```

```
Private Sub cmdStart_Click()
```

```
    Call Biegung_Berechnung
```

```
End Sub
```

```
Private Sub cmdTest_Click()
```

```
    TextBox1 = 40
```

```
    TextBox2 = 2
```

```
    TextBox3 = 10000
```

```
    TextBox4 = 40000
```

```
    TextBox5 = 2100
```

```
    TextBox6 = 600000
```

```
End Sub
```

```
Sub Biegung_Berechnung()
```

```
    Dim Lg, Fe, Qs, Em, Ia, dx, x As Double
```

```
    Dim Mx, xl, yx As Double
```

```
    Dim i As Integer
```

```
    Dim Zl As String
```

```
    Lg = Val(TextBox1)
```

```
    dx = Val(TextBox2)
```

```
    Fe = Val(TextBox3)
```

```
    Qs = Val(TextBox4)
```

```
    Em = Val(TextBox5)
```

```
    Ia = Val(TextBox6)
```

```
    If Lg = 0 Or dx = 0 Or Fe = 0 Or Qs = 0 Or _
```

```
        Em = 0 Or Ia = 0 Then
```

```
        MsgBox "Fehlerhafte Dateneingabe!", _
```

```
            vbCritical + vbOKOnly, "ACHTUNG"
```

```
        Exit Sub
```

```
    End If
```

```
    Worksheets("Biegeträger").Activate
```

```
    Worksheets("Biegeträger").Cells.Clear
```

```
    Range("A1") = "Stelle" & vbLf & "x [cm]"
```

```
    Range("B1") = "Moment" & vbLf & "M [Ncm]"
```

```
    Range("C1") = "Biegung" & vbLf & "x [cm]"
```

```
    Columns("A:C").EntireColumn.AutoFit
```

```
    i = 1
```

```
    For x = 0 To Lg Step dx
```

```
        xl = x / Lg
```

```
        Mx = Lg / 2 * (1 - xl) * (2 * Fe + Qs * (1 - xl))
```

```
        yx = -Lg ^ 3 / (24 * Em * Ia) * _
```

```
            (4 * Fe * (3 * xl ^ 2 - xl ^ 3) + _
```

```
            Qs * (6 * xl ^ 2 - 4 * xl ^ 3 + xl ^ 4))
```

```
        i = i + 1
```

```
        Zl = Right("00" + LTrim(Str(i)), 2)
```

```
        Range("A" + Zl) = x
```

```
        Range("B" + Zl) = Round(Mx, 3)
```

```
        Range("C" + Zl) = Round(yx, 6)
```

```
    Next x
```

```
    Unload Me
```

```
End Sub
```

```

Option Explicit
'Prozedur zur Erstellung eines Formblatts
Sub Formblatt()
'
'Tabelle löschen
Worksheets("Massenträgheitsmomente").Activate
Worksheets("Massenträgheitsmomente").Cells.Clear
'
'Formblatt
Range("A1") = "Form"
Range("B1:D1").MergeCells = True
Range("B1") = "Maße"
Range("C1") = "Maße"
Range("D1") = "Maße"
Range("B2") = "a/R [mm]"
Range("C2") = "b/r[mm]"
Range("D2") = "h [mm]"
Range("E1") = "Dichte"
Range("E2") = "[kg/dm" + Chr(179) + "]"
Range("F1") = "Masse m"
Range("F2") = "[kg]"
Range("G1") = "Moment Id"
Range("G2") = "[kgm" + Chr(178) + "]"
Range("H1") = "Abstand x"
Range("H2") = "[mm]"
Range("I1") = "Moment Ix"
Range("I2") = "[kgm" + Chr(178) + "]"
Range("J1") = "Gesamt Ix"
Range("J2") = "[kgm" + Chr(178) + "]"
Range("A:F").ColumnWidth = 10
Range("G:G").ColumnWidth = 20
Range("H:H").ColumnWidth = 10
Range("I:J").ColumnWidth = 20
Columns("A:J").Select
Selection.NumberFormat = "0.00"
Zeile = 3
Gesamt = 0
Zelle
End Sub

Sub Quader()
Load frmQuader
frmQuader.Show
End Sub

Sub Zylinder()
Load frmZylinder
frmZylinder.Show
End Sub

Sub Kugel()
Load frmKugel
frmKugel.Show
End Sub

Sub Kegel()
Load frmKegel
frmKegel.Show
End Sub

Sub Ring()
Load frmRing
frmRing.Show
End Sub

```

```
Option Explicit
Private Sub cmdKegel_Click()
    Kegel
End Sub

Private Sub UserForm_Activate()
    TextBox1.SetFocus
End Sub

Sub Kegel()
    Dim r1, r2, h, d, x As Double
    Dim m, Id, Ix As Double
    r1 = Val(TextBox1)
    r2 = Val(TextBox2)
    h = Val(TextBox3)
    d = Val(TextBox4)
    x = Val(TextBox5)
    m = 3.1415926 / 3 * h * (r1 * r1 + r1 * r2 + r2 * r2) / 1000000 * d
    Id = 0.3 * m * (r1 ^ 5 - r2 ^ 5) / (r1 ^ 3 - r2 ^ 3) / 1000000
    Ix = Id + m * x * x / 1000000
    Gesamt = Gesamt + Ix
    If Zeile = 0 Then Zeile = 3
    Cells(Zeile, 1) = "Kegel"
    Cells(Zeile, 2) = r1
    Cells(Zeile, 3) = r2
    Cells(Zeile, 4) = h
    Cells(Zeile, 5) = d
    Cells(Zeile, 6) = m
    Cells(Zeile, 7) = Id
    Cells(Zeile, 8) = x
    Cells(Zeile, 9) = Ix
    Cells(Zeile, 10) = Gesamt
    Zeile = Zeile + 1
    Zelle
    Unload Me
End Sub
```

```
Option Explicit
Private Sub cmdKugel_Click()
    Kugel
End Sub

Private Sub UserForm_Activate()
    TextBox1.SetFocus
End Sub

Sub Kugel()
    Dim r1, r2, d, x As Double
    Dim m, Id, Ix As Double
    r1 = Val(TextBox1)
    r2 = Val(TextBox2)
    d = Val(TextBox4)
    x = Val(TextBox5)
    m = (r1 ^ 3 - r2 ^ 3) * 4 / 3 * 3.1415926 / 1000000 * d
    Id = 0.4 * m * (r1 ^ 5 - r2 ^ 5) / (r1 ^ 3 - r2 ^ 3) / 1000000
    Ix = Id + m * x * x / 1000000
    Gesamt = Gesamt + Ix
    If Zeile = 0 Then Zeile = 3
    Cells(Zeile, 1) = "Kugel"
    Cells(Zeile, 2) = r1
    Cells(Zeile, 3) = r2
    Cells(Zeile, 4) = ""
    Cells(Zeile, 5) = d
    Cells(Zeile, 6) = m
    Cells(Zeile, 7) = Id
    Cells(Zeile, 8) = x
    Cells(Zeile, 9) = Ix
    Cells(Zeile, 10) = Gesamt
    Zeile = Zeile + 1
    Zelle
    Unload Me
End Sub
```

Option Explicit

```
Private Sub cmdQuader_Click()  
    Quader  
End Sub
```

```
Private Sub UserForm_Activate()  
    TextBox1.SetFocus  
End Sub
```

```
Sub Quader()  
    Dim a, b, h, d, x As Double  
    Dim m, Id, Ix As Double  
    a = Val(TextBox1)  
    b = Val(TextBox2)  
    h = Val(TextBox3)  
    d = Val(TextBox4)  
    x = Val(TextBox5)  
    m = a * b * h / 1000000 * d  
    Id = m / 12 * (a * a + b * b) / 1000000  
    Ix = Id + m * x * x / 1000000  
    Gesamt = Gesamt + Ix  
    If Zeile = 0 Then Zeile = 3  
    Cells(Zeile, 1) = "Quader"  
    Cells(Zeile, 2) = a  
    Cells(Zeile, 3) = b  
    Cells(Zeile, 4) = h  
    Cells(Zeile, 5) = d  
    Cells(Zeile, 6) = m  
    Cells(Zeile, 7) = Id  
    Cells(Zeile, 8) = x  
    Cells(Zeile, 9) = Ix  
    Cells(Zeile, 10) = Gesamt  
    Zeile = Zeile + 1  
    Zelle  
    Unload Me  
End Sub
```

```
Option Explicit
Private Sub cmdRing_Click()
    Ring
End Sub

Private Sub UserForm_Activate()
    TextBox1.SetFocus
End Sub

Sub Ring()
    Dim r1, r2, d, x As Double
    Dim m, Id, Ix As Double
    r1 = Val(TextBox1)
    r2 = Val(TextBox2)
    d = Val(TextBox4)
    x = Val(TextBox5)
    m = 2 * 3.1415926 ^ 2 * r2 * r2 * r1 / 1000000 * d
    Id = m * (r1 * r1 + 3 / 4 * r2 * r2) / 1000000
    Ix = Id + m * x * x / 1000000
    Gesamt = Gesamt + Ix
    If Zeile = 0 Then Zeile = 3
    Cells(Zeile, 1) = "Ring"
    Cells(Zeile, 2) = r1
    Cells(Zeile, 3) = r2
    Cells(Zeile, 4) = ""
    Cells(Zeile, 5) = d
    Cells(Zeile, 6) = m
    Cells(Zeile, 7) = Id
    Cells(Zeile, 8) = x
    Cells(Zeile, 9) = Ix
    Cells(Zeile, 10) = Gesamt
    Zeile = Zeile + 1
    Zelle
    Unload Me
End Sub
```

```
Option Explicit
Private Sub cmdRing_Click()
    Ring
End Sub

Private Sub UserForm_Activate()
    TextBox1.SetFocus
End Sub

Sub Ring()
    Dim r1, r2, d, x As Double
    Dim m, Id, Ix As Double
    r1 = Val(TextBox1)
    r2 = Val(TextBox2)
    d = Val(TextBox4)
    x = Val(TextBox5)
    m = 2 * 3.1415926 ^ 2 * r2 * r2 * r1 / 1000000 * d
    Id = m * (r1 * r1 + 3 / 4 * r2 * r2) / 1000000
    Ix = Id + m * x * x / 1000000
    Gesamt = Gesamt + Ix
    If Zeile = 0 Then Zeile = 3
    Cells(Zeile, 1) = "Ring"
    Cells(Zeile, 2) = r1
    Cells(Zeile, 3) = r2
    Cells(Zeile, 4) = ""
    Cells(Zeile, 5) = d
    Cells(Zeile, 6) = m
    Cells(Zeile, 7) = Id
    Cells(Zeile, 8) = x
    Cells(Zeile, 9) = Ix
    Cells(Zeile, 10) = Gesamt
    Zeile = Zeile + 1
    Calle Zelle
    Unload Me
End Sub
```

```
Option Explicit
Public Zeile As Integer
Public Gesamt As Double
Sub Zelle()
    Dim z1 As String
    z1 = LTrim(Str(Zeile))
    Range("A" + z1).Select
End Sub
```

```

Option Explicit
'
'Prozedur zur Erstellung eines Formblatts
Sub Formblatt()
'
'Tabelle löschen
Worksheets("Schwingung").Activate
Worksheets("Schwingung").Cells.Clear
'
'Formblatt
Range("A:A").ColumnWidth = 30
Range("A1") = "Masse m [kg]"
Range("A2") = "Federkonstante f [kg/s" + ChrW(178) + "]"
Range("A3") = "Dämpfungskonstante d [1/s]"
Range("A4") = "Ausgangsposition s0 [m]"
Range("A5") = "Ausgangsgeschwindigkeit v0 [m/s]"
Range("A6") = "Ausgangszeit t [s]"
Range("A7") = "Schrittweite dt [s]"
Range("A8") = "Endzeit te [s]"
Range("B:E").ColumnWidth = 10
Range("C1:E1").MergeCells = True
Range("C1") = "Auswertung"
Range("C2") = "t [s]"
Range("D2") = "v [m/s]"
Range("E2") = "s [m]"
Columns("B:E").Select
Selection.NumberFormat = "0.00"
Range("B1").Select
End Sub

Sub Testdaten()
Cells(1, 2) = 50
Cells(2, 2) = 80
Cells(3, 2) = 0.4
Cells(4, 2) = -5
Cells(5, 2) = 0
Cells(6, 2) = 0
Cells(7, 2) = 0.1
Cells(8, 2) = 10
End Sub

Sub Simulation()
Dim m, f, d, s, v, t, dt, te As Double
Dim dV, ds As Double
Dim i As Integer
m = Cells(1, 2)
f = Cells(2, 2)
d = Cells(3, 2)
s = Cells(4, 2)
v = Cells(5, 2)
t = Cells(6, 2)
dt = Cells(7, 2)
te = Cells(8, 2)
i = 2
Do
    dV = -(f / m * s + 2 * d * v) * dt
    v = v + dV
    ds = v * dt
    s = s + ds
    t = t + dt
    i = i + 1
    Cells(i, 3) = t
    Cells(i, 4) = v
    Cells(i, 5) = s
Loop While t < te
End Sub

Public Sub Schwingung_Zeigen()
Dim MyDoc As Object
Dim xlRange As Range
Dim lngNumRows, lngNumCols As Long
'
'Verweis auf Worksheet mit Daten
Set MyDoc = ThisWorkbook.Worksheets("Schwingung")
'
'Übergabe der Anzahl der Spalten/Zeilen:
lngNumRows = MyDoc.UsedRange.Rows.Count

```

```

    lngNumCols = MyDoc.UsedRange.Columns.Count
,
'Verweis auf Datenbereich setzen:
    Set xlRange = MyDoc.Range("E3:E" + LTrim(Str(lngNumRows)))
,
'Diagramm erstellen:
    CreateChartObjectRange xlRange, MyDoc
,
'Verweise freigeben:
    Set xlRange = Nothing
    Set MyDoc = Nothing
End Sub

Public Sub CreateChartObjectRange(ByVal xlRange As Range, ByVal MyDoc As Object)
    Dim objChart As Object
,
'Bildschirmaktualisierung deaktivieren:
    Application.ScreenUpdating = False
,
'Verweis auf Diagramm setzen und Diagramm hinzufügen:
    Set objChart = Application.Charts.Add
    With objChart
        'Diagramm-Typ und -Quelldatenbereich festlegen:
        .ChartType = xlLineStacked
        .SetSourceData Source:=xlRange, PlotBy:=xlColumns
        'Titel festlegen:
        .HasTitle = True
        .ChartTitle.Text = "Freie gedämpfte Schwingung"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Zeit [s]"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Weg [m]"
        'Diagramm auf Tabellenblatt einbetten:
        .Location Where:=xlLocationAsObject, Name:=MyDoc.Name
    End With
,
'Legende löschen
    ActiveChart.Legend.Select
    Selection.Delete
,
'Verweis auf das eingebettete Diagramm setzen:
    Set objChart = MyDoc.ChartObjects(MyDoc.ChartObjects.Count)
    With objChart
        .Left = 400
        .Top = 50
        .Width = 300
        .Height = 200
    End With
,
'Bildschirmaktualisierung aktivieren:
    Application.ScreenUpdating = True
    Set objChart = Nothing
End Sub

Sub Schwingung_Loeschen()
    Dim MyDoc As Worksheet
    Set MyDoc = ThisWorkbook.Worksheets("Schwingung")
    Dim Shp As Shape
,
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
End Sub

```

```

Option Explicit
'Prozedur zur Erstellung eines Formblatts
Sub Formblatt()
'
'Tabelle löschen
Worksheets("Geschweißter Biegeträger").Activate
Worksheets("Geschweißter Biegeträger").Cells.Clear
'
'Formblatt
Range("A1") = "FN [N]"
Range("A2") = "FQ [N]"
Range("A3") = "Mb [Nm]"
Range("A5") = "H [mm]"
Range("A6") = "e [mm]"
Range("A7") = "A [cm" + ChrW(178) + "]"
Range("A8") = "I [cm^4]"
Range("A9") = ChrW(963) & "z [N/cm" + ChrW(178) + "]"
Range("A10") = ChrW(964) & "a [N/cm" + ChrW(178) + "]"
Range("A11") = ChrW(963) & "b1 [N/cm" + ChrW(178) + "]"
Range("A12") = ChrW(963) & "b2 [N/cm" + ChrW(178) + "]"
Range("A13") = ChrW(963) & "h1 [N/cm" + ChrW(178) + "]"
Range("A14") = ChrW(963) & "h2 [N/cm" + ChrW(178) + "]"

Range("C1:E1").MergeCells = True
Range("C1:E1") = "Maße"
Range("C2") = "b [mm]"
Range("D2") = "h [mm]"
Range("E2") = "a [mm]"

Range("F1") = "Schwerpkt."
Range("F2") = "e [mm]"
Range("G1:I1").MergeCells = True
Range("G1:I1") = "Schweißnähte"
Range("G2") = ChrW(963) & "b [N/cm" + ChrW(178) + "]"
Range("H2") = ChrW(964) & "a [N/cm" + ChrW(178) + "]"
Range("I2") = ChrW(963) & "h [N/cm" + ChrW(178) + "]"

Range("A:A").ColumnWidth = 12
Range("B:I").ColumnWidth = 10
Columns("A:I").Select
Selection.NumberFormat = "0.00"
Range("A2").Select
End Sub

Sub Testbeispiel()
Cells(1, 2) = 5000
Cells(2, 2) = 1000
Cells(3, 2) = 500
Cells(3, 3) = 600
Cells(3, 4) = 100
Cells(3, 5) = 0
Cells(4, 3) = 200
Cells(4, 4) = 200
Cells(4, 5) = 5
Cells(5, 3) = 100
Cells(5, 4) = 100
Cells(5, 5) = 4
Cells(6, 3) = 400
Cells(6, 4) = 50
Cells(6, 5) = 4
End Sub

Sub Auswertung()
Dim Zeile, i As Integer
Dim H, e, si As Double
Dim s1, s2, s3, s4 As Double 'Summen
s1 = 0: s2 = 0: s3 = 0: s4 = 0
Zeile = 3
'
'Summenbildung
Do While Cells(Zeile, 3) > 0
e = H + Cells(Zeile, 4) / 2
Cells(Zeile, 6) = e
H = H + Cells(Zeile, 4)
s1 = s1 + Cells(Zeile, 3) * Cells(Zeile, 4)
s2 = s2 + Cells(Zeile, 3) * Cells(Zeile, 4) * e

```



```

Option Explicit
'Prozedur zur Erstellung eines Formblatts
Sub Formblatt()
'
'Tabelle löschen
  Worksheets("Monte-Carlo").Activate
  Worksheets("Monte-Carlo").Cells.Clear
'
'Formblatt
  Range("A1") = "F [N]"
  Range("A2") = "E [N/cm" + ChrW(178) + "]"
  Range("A3") = "k "
  Range("A4") = ChrW(963) & "zul [N/cm" + ChrW(178) + "]"
  Range("A5") = "fzul [cm]"
  Range("A6") = "hmin [cm]"
  Range("A7") = "hmax [cm]"
  Range("A8") = "lmin [cm]"
  Range("A9") = "lmax [cm]"
  Range("A10") = "n"
  Range("C1:I1").MergeCells = True
  Range("C1:I1") = "Auswertung"
  Range("C2") = "n"
  Range("D2") = "h [cm]"
  Range("E2") = "l [cm]"
  Range("F2") = ChrW(963) & " [N/cm" + ChrW(178) + "]"
  Range("G2") = "I [cm^4]"
  Range("H2") = "f [cm]"
  Range("I2") = "V [cm" + ChrW(179) + "]"
  Range("A:B").ColumnWidth = 15
  Range("C:I").ColumnWidth = 10
  Columns("A:G").Select
  Selection.NumberFormat = "0.00"
  Columns("H:H").Select
  Selection.NumberFormat = "0.00000"
  Columns("I:I").Select
  Selection.NumberFormat = "0.000"
  Range("A2").Select
End Sub

Sub Testdaten()
  Cells(1, 2) = 20000
  Cells(2, 2) = 21000000
  Cells(3, 2) = 0.25
  Cells(4, 2) = 6000
  Cells(5, 2) = 0.3
  Cells(6, 2) = 22
  Cells(7, 2) = 26
  Cells(8, 2) = 150
  Cells(9, 2) = 250
  Cells(10, 2) = 1000000
End Sub

Sub Auswertung()
  Dim F, E, l, k, Szul, fzul, hmin, hmax, lmin, lmax, n As Double
  Dim i, j As Integer
  Dim x, V0, hi, li, Si, Ii, Vi, fi As Double
  F = Cells(1, 2)
  E = Cells(2, 2)
  k = Cells(3, 2)
  Szul = Cells(4, 2)
  fzul = Cells(5, 2)
  hmin = Cells(6, 2)
  hmax = Cells(7, 2)
  lmin = Cells(8, 2)
  lmax = Cells(9, 2)
  n = Cells(10, 2)

  V0 = k * hmax * hmax * lmax
  i = 0: j = 2
  Randomize
  x = 0
  Do
    x = Rnd(x)
    hi = (hmax - hmin) * x + hmin
    x = Rnd(x)
    li = (lmax - lmin) * x + lmin
    Si = 6 * F * li / (k * hi ^ 3)
    If Si <= Szul Then

```

```
Ii = hi ^ 4 / 48
fi = F * li ^ 3 / (3 * E * Ii)
If fi <= fzul Then
  Vi = k * hi ^ 2 * li
  If Vi <= V0 Then
    j = j + 1
    Cells(j, 3) = i
    Cells(j, 4) = hi
    Cells(j, 5) = li
    Cells(j, 6) = Si
    Cells(j, 7) = Ii
    Cells(j, 8) = fi
    Cells(j, 9) = Vi
    V0 = Vi
  End If
End If
End If
End If
i = i + 1
Loop While i <= n
End Sub
```

```
Option Explicit
Private Sub Worksheet_Change(ByVal Target As Range)
    Dim Zeile As Integer
    Dim Zl As String
    If ActiveCell.Column = 2 Then
        Zeile = ActiveCell.Row
        If Zeile > 0 Then
            Zeile = Zeile - 1
            Zl = Right("00" + LTrim(Str(Zeile)), 2)
            Range("C" + Zl).Activate
            ActiveCell.ClearComments
            Select Case Target.Text
            Case "R"
                Load frmRechteck
                frmRechteck.Show
            Case "D"
                Load frmDreieck
                frmDreieck.Show
            Case "Z"
                Load frmZylinder
                frmZylinder.Show
            Case "S"
                Call Volumen_Summe
            Case ""
            Case Else
                MsgBox "Unzulässige Eingabe !"
                Zl = Right("00" + LTrim(Str(Zeile + 1)), 2)
                Range("B" + Zl).Activate
            End Select
        End If
    End If
End Sub
```

```
Sub Volumen_Summe()
    Dim i, Zeile As Integer
    Dim Zl As String
    Dim w, Sum As Double
    Zeile = ActiveCell.Row
    Sum = 0
    For i = 4 To Zeile - 1
        Zl = Right("00" + LTrim(Str(i)), 2)
        w = CDBl(Range("H" + Zl).Value)
        Sum = Sum + w
    Next i
    Zl = Right("00" + LTrim(Str(Zeile)), 2)
    Range("H" + Zl).Value = Round(Sum, 3)
    Range("B" + Zl + ":" + "H" + Zl).Interior.ColorIndex = 6
    Call VolumenKommentar_Aus
    Range("H" + Zl).Activate
End Sub
```

```
Private Sub TextBox4_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    If KeyCode = 13 Then
        Call Rechteckplatte
        Unload Me
    End If
End Sub
Sub Rechteckplatte()
    Dim Stk, Zeile As Integer
    Dim a, b, c, Vol As Double
    Zeile = ActiveCell.Row
    Zl$ = Right("00" + LTrim(Str(Zeile)), 2)
    Stk = Val(TextBox1)
    a = Val(TextBox2)
    b = Val(TextBox3)
    c = Val(TextBox4)
    Vol = Stk * a * b * c / 1000000
    Range("C" + Zl$).Value = Stk
    Range("D" + Zl$).Value = a
    Range("E" + Zl$).Value = b
    Range("F" + Zl$).Value = c
    Range("G" + Zl$).Value = " "
    Range("H" + Zl$).Value = Round(Vol, 3)
    Zl$ = Right("00" + LTrim(Str(Zeile + 1)), 2)
    Range("B" + Zl$).Activate
End Sub
Private Sub UserForm_Activate()
    Zeile = ActiveCell.Row
    Zl$ = Right("00" + LTrim(Str(Zeile)), 2)
    TextBox1 = Range("C" + Zl$).Value
    TextBox2 = Range("D" + Zl$).Value
    TextBox3 = Range("E" + Zl$).Value
    TextBox4 = Range("F" + Zl$).Value
End Sub
```

```
Private Sub cmdRechtw_Click()  
    a = Val(TextBox2)  
    b = Val(TextBox3)  
    c = Sqr(a * a + b * b)  
    TextBox4 = Round(c, 3)  
    TextBox4.SetFocus  
End Sub  
  
Private Sub TextBox5_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)  
    If KeyCode = 13 Then  
        Call Dreieckplatte  
        Unload Me  
    End If  
End Sub  
  
Sub Dreieckplatte()  
    Dim Stk, Zeile As Integer  
    Dim a, b, c, d, E, F, g, Vol As Double  
    Zeile = ActiveCell.Row  
    Zl$ = Right("00" + LTrim(Str(Zeile)), 2)  
    Stk = Val(TextBox1)  
    a = Val(TextBox2)  
    b = Val(TextBox3)  
    c = Val(TextBox4)  
    d = Val(TextBox5)  
    E = (a + b + c) / 2  
    F = E * (E - a) * (E - b) * (E - c)  
    g = Sqr(F)  
    Vol = Stk * g / 1000000  
    Range("C" + Zl$).Value = Stk  
    Range("D" + Zl$).Value = a  
    Range("E" + Zl$).Value = b  
    Range("F" + Zl$).Value = c  
    Range("G" + Zl$).Value = d  
    Range("H" + Zl$).Value = Round(Vol, 3)  
    Zl$ = Right("00" + LTrim(Str(Zeile + 1)), 2)  
    Range("B" + Zl$).Activate  
End Sub  
  
Private Sub UserForm_Activate()  
    Zeile = ActiveCell.Row  
    Zl$ = Right("00" + LTrim(Str(Zeile)), 2)  
    TextBox1 = Range("C" + Zl$).Value  
    TextBox2 = Range("D" + Zl$).Value  
    TextBox3 = Range("E" + Zl$).Value  
    TextBox4 = Range("F" + Zl$).Value  
    TextBox5 = Range("G" + Zl$).Value  
End Sub
```

```
Private Sub TextBox3_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, ByVal Shift As Integer)
    If KeyCode = 13 Then
        Call Zylinder
        Unload Me
    End If
End Sub
```

```
Sub Zylinder()
    Dim Stk, Zeile As Integer
    Dim a, b, Vol As Double
    Zeile = ActiveCell.Row
    Zl$ = Right("00" + LTrim(Str(Zeile)), 2)
    Stk = Val(TextBox1)
    a = Val(TextBox2)
    b = Val(TextBox3)
    Vol = Stk * a * a * 3.14159 / 4 / 1000000
    Range("C" + Zl$).Value = Stk
    Range("D" + Zl$).Value = a
    Range("E" + Zl$).Value = b
    Range("F" + Zl$).Value = c
    Range("G" + Zl$).Value = " "
    Range("H" + Zl$).Value = Round(Vol, 3)
    Zl$ = Right("00" + LTrim(Str(Zeile + 1)), 2)
    Range("B" + Zl$).Activate
End Sub
```

```
Private Sub UserForm_Activate()
    Zeile = ActiveCell.Row
    Zl$ = Right("00" + LTrim(Str(Zeile)), 2)
    TextBox1 = Range("C" + Zl$).Value
    TextBox2 = Range("D" + Zl$).Value
    TextBox3 = Range("E" + Zl$).Value
    TextBox4 = Range("F" + Zl$).Value
    TextBox5 = Range("G" + Zl$).Value
End Sub
```

```
Option Explicit
Sub Volumenformblatt_Neu()
  Dim i, x, y As Integer
  Dim Zl, Sp As String
  Worksheets("Volumen").Activate
  Worksheets("Volumen").Cells.Clear
  Range("B2:G2").MergeCells = True
  Range("B2:G2") = "Volumenberechnung"
  Range("B3") = "Form"
  Range("C3") = "Anzahl" + vbLf + "+/-"
  Range("D3") = "Maß a" + vbLf + "[mm]"
  Range("E3") = "Maß b" + vbLf + "[mm]"
  Range("F3") = "Maß c" + vbLf + "[mm]"
  Range("G3") = "Maß d" + vbLf + "[mm]"
  Range("H3") = "Volumen" + vbLf + "[dm" + Chr(179) + "]"
  Call Volumen_Kommentar_Ein
End Sub
```

```
Sub Volumen_Kommentar_Ein()
  Dim tx As String
  tx = "In dieser Spalte:" + vbLf
  tx = tx + "R für Rechteckplatte," + vbLf
  tx = tx + "D für Dreieckplatte," + vbLf
  tx = tx + "Z für Zylinder" + vbLf
  tx = tx + "S für Summenbildung" + vbLf
  tx = tx + "eingeben!"
  Range("B4").Activate
  With ActiveCell
    .AddComment tx
    .Comment.Visible = True
  End With
End Sub
```

```
Sub VolumenKommentar_Aus()
  Range("B4").Activate
  With ActiveCell
    .ClearComments
  End With
End Sub
```

```

Option Explicit
Sub Formblatt()
    Dim i, x, y As Integer
    Dim Zl, Sp As String
    Worksheets("Durchbiegung").Activate
    Worksheets("Durchbiegung").Cells.Clear
    Range("A1:B1").MergeCells = True
    Range("A1") = "E-Modul [N/mm" + ChrW(178) + "]" = "
    Range("E1:F1").MergeCells = True
    Range("E1") = "Schrittweite " + ChrW(8710) + "x [mm]="

    Range("A2:B2").MergeCells = True
    Range("A2:B2") = "Einzelkräfte"
    Range("A3") = "F" + vbLf + "[N]"
    Range("B3") = "l" + vbLf + "[mm]"
    Range("C2:D2").MergeCells = True
    Range("C2:D2") = "Einzelmomente"
    Range("C3") = "Mb" + vbLf + "[Nmm]"
    Range("D3") = "l" + vbLf + "[mm]"
    Range("E2:G2").MergeCells = True
    Range("E2:G2") = "Streckenlasten"
    Range("E3") = "q" + vbLf + "[N/mm]"
    Range("F3") = "l1" + vbLf + "[mm]"
    Range("G3") = "l2" + vbLf + "[mm]"
    Range("H2:I2").MergeCells = True
    Range("H2:I2") = "Abmessungen"
    Range("H3") = "D" + vbLf + "[mm]"
    Range("I3") = "l" + vbLf + "[mm]"
    Range("J3") = "I" + vbLf + "[cm^4]"
    Range("K2:M2").MergeCells = True

    Range("K2:M2") = "Auswertung"
    Range("K3") = "x" + vbLf + "[mm]"
    Range("L3") = ChrW(8710) + ChrW(945) + vbLf
    Range("M3") = ChrW(8710) + "f" + vbLf + "[mm]"
    Range("A4").Select
End Sub

```

```

Sub Testdaten_Links()
    Cells(1, 3) = 210000
    Cells(1, 7) = 10
    Cells(4, 1) = 6000
    Cells(4, 2) = 1200
    Cells(4, 5) = 2.4
    Cells(4, 6) = 0
    Cells(4, 7) = 900
    Cells(5, 5) = 2
    Cells(5, 6) = 900
    Cells(5, 7) = 1500
    Cells(6, 5) = 1.6
    Cells(6, 6) = 1500
    Cells(6, 7) = 1900
    Cells(4, 8) = 96
    Cells(4, 9) = 900
    Cells(5, 8) = 90
    Cells(5, 9) = 1500
    Cells(6, 8) = 78
    Cells(6, 9) = 1900
End Sub

```

```

Sub Testdaten_Rechts()
    Cells(1, 3) = 210000
    Cells(1, 7) = 10
    Cells(4, 1) = 6000
    Cells(4, 2) = 1200
    Cells(4, 5) = 2.4
    Cells(4, 6) = 0
    Cells(4, 7) = 900
    Cells(5, 5) = 2
    Cells(5, 6) = 900
    Cells(5, 7) = 1700
    Cells(6, 5) = 1.6
    Cells(6, 6) = 1700
    Cells(6, 7) = 2100
    Cells(4, 8) = 96
    Cells(4, 9) = 900
    Cells(5, 8) = 90

```

```

Cells(5, 9) = 1700
Cells(6, 8) = 78
Cells(6, 9) = 2100
End Sub

Sub Auswertung()
Dim x, M0, M1, Fi, li, Mi, l1i, l2i, qi As Double
Dim E, dx, lmax, Di, Ii, y, da, df, Sf As Double
Dim i, k, m, n As Integer
x = 0: n = 0
M0 = 0: M1 = 0
E = Cells(1, 3)
dx = Cells(1, 7)
Sf = 0
,
'Auswertungsschleife
Do
,
'Einzelkräfte
i = 0
Do
i = i + 1
Fi = Cells(3 + i, 1)
li = Cells(3 + i, 2)
If Fi > 0 Then
If x < li Then
M1 = M1 + Fi * (li - x)
End If
End If
Loop While Fi > 0
,
'Einzelmomente
i = 0
Do
i = i + 1
Mi = Cells(3 + i, 3)
li = Cells(3 + i, 4)
If Mi > 0 Then
If x < li Then
M1 = M1 + Mi
End If
End If
Loop While Mi > 0
,
'Streckenlasten
i = 0
Do
i = i + 1
qi = Cells(3 + i, 5)
l1i = Cells(3 + i, 6)
l2i = Cells(3 + i, 7)
If qi > 0 Then
If x < l2i Then
If x < l1i Then
M1 = M1 + qi * (l2i - l1i) * ((l1i + l2i) / 2 - x)
Else
M1 = M1 + qi * (l2i - x) * ((x + l2i) / 2 - x)
End If
End If
End If
Loop While qi > 0
,
'Flächenträgheitsmoment
i = 0
Ii = 0
Do
i = i + 1
Di = Cells(3 + i, 8)
li = Cells(3 + i, 9)
If Di > 0 Then
If x < li Then
If Ii = 0 Then
Ii = 3.1415926 / 4 * (Di / 2) ^ 4
Cells(3 + i, 10) = Ii
End If
End If
lmax = li

```

```

    End If
    Loop While Di > 0
,
'schrittweite
    If x > 0 Then
        Mi = (M0 + M1) / 2
        da = Mi * dx / Ii / E
        df = dx / 2 * da
        Sf = Sf + df
        n = n + 1
        Cells(n + 3, 11) = x
        Cells(n + 3, 12) = da
        Cells(n + 3, 13) = Sf
    End If
    x = x + dx
    M0 = M1
,
'Schleifenende
    Loop While x < lmax
End Sub
,
'Darstellung der Durchbiegung als Diagramm
Sub Durchbiegung_Zeigen()
    Dim xlRange As Range
    Dim lngNumRows As Long
    Dim lngNumCols As Long
,
'Verweis auf Worksheet mit Daten
    Set objLinienDiagramm = ThisWorkbook.Worksheets("Durchbiegung")
,
'Übergabe der Anzahl der Spalten/Zeilen:
    lngNumRows = objLinienDiagramm.UsedRange.Rows.Count
    lngNumCols = objLinienDiagramm.UsedRange.Columns.Count
,
'Verweis auf Datenbereich setzen:
    Set xlRange = objLinienDiagramm.Range("M4:M" + LTrim(Str(lngNumRows)))
,
'Diagramm erstellen:
    intLeft = 100
    intTop = 100
    intWidth = 300
    intHeight = 200
    DTitel = "Durchbiegung"
    xTitel = "x [mm] "
    yTitel = "y [mm]"
    CreateChartObjectRange xlRange
,
'Verweise freigeben:
    Set xlRange = Nothing
    Set objLinienDiagramm = Nothing
End Sub
,
'Darstellung der Winkeländerung als Diagramm
Sub Winkeländerung_Zeigen()
    Dim xlRange As Range
    Dim lngNumRows As Long
    Dim lngNumCols As Long
,
'Verweis auf Worksheet mit Daten
    Set objLinienDiagramm = ThisWorkbook.Worksheets("Durchbiegung")
,
'Übergabe der Anzahl der Spalten/Zeilen:
    lngNumRows = objLinienDiagramm.UsedRange.Rows.Count
    lngNumCols = objLinienDiagramm.UsedRange.Columns.Count
,
'Verweis auf Datenbereich setzen:
    Set xlRange = objLinienDiagramm.Range("L4:L" + LTrim(Str(lngNumRows)))
,
'Diagramm erstellen:
    intLeft = 200
    intTop = 150
    intWidth = 300
    intHeight = 200
    DTitel = "Winkeländerung"
    xTitel = "x [mm] "
    yTitel = "alpha"
    CreateChartObjectRange xlRange

```

```
'Verweise freigeben:  
  Set xlRange = Nothing  
  Set objLinienDiagramm = Nothing  
End Sub  
Sub Durchbiegung_Löschen()  
'  
'Verweis auf Worksheet mit Daten  
  Set objLinienDiagramm = ThisWorkbook.Worksheets("Durchbiegung")  
  LinienDiagramme_Löschen  
End Sub
```

```
Option Explicit
Public objLinienDiagramm As Object 'als Worksheet
Public DTitel, xTitel, yTitel As String
Public intLeft, intTop, intWidth, intHeight As Integer
'
'Erstellung eines Linien-Diagramms
Public Sub CreateChartObjectRange(ByVal xlRange As Range)
    Dim objChart As Object
'
'Bildschirmaktualisierung deaktivieren:
    Application.ScreenUpdating = False
'
'Verweis auf Diagramm setzen und Diagramm hinzufügen:
    Set objChart = Application.Charts.Add
    With objChart
        'Diagramm-Typ und -Quelldatenbereich festlegen:
        .ChartType = xlLine
        .SetSourceData Source:=xlRange, PlotBy:=xlColumns
        'Titel festlegen:
        .HasTitle = True
        .ChartTitle.Text = DTitel
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = xTitel
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = yTitel
        'Diagramm auf Tabellenblatt einbetten:
        .Location Where:=xlLocationAsObject, Name:=objLinienDiagramm.Name
    End With
'
'Legende löschen
    ActiveChart.Legend.Select
    Selection.Delete
'
'Verweis auf das eingebettete Diagramm setzen:
    Set objChart = _
        objLinienDiagramm.ChartObjects(objLinienDiagramm.ChartObjects.Count)
    With objChart
        .Left = intLeft
        .Top = intTop
        .Width = intWidth
        .Height = intHeight
    End With
'
'Bildschirmaktualisierung aktivieren:
    Application.ScreenUpdating = True
    Set objChart = Nothing
End Sub
'
'Entfernung aller Shapes aus der Tabelle
Sub LinienDiagramme_Löschen()
    Dim Shp As Shape
'
'alle Charts löschen
    For Each Shp In objLinienDiagramm.Shapes
        Shp.Delete
    Next
End Sub
```

```
Option Explicit
Sub Transponieren()
    Dim i, j, k As Integer
    ,
'Ermittelt die Anzahl Werte in der Spalte C
    i = 4
    Do
        If Cells(i, 3) > 0 Then j = i
        i = i + 1
    Loop While Cells(i, 3) <> 0
    ,
'Tranponiert die Werte aus C nach I
'in umgekehrter Reihenfolge
    k = 3
    For i = j To 4 Step -1
        k = k + 1
        Cells(k, 9) = Cells(i, 3)
    Next i
    ,
'Ermittelt die Anzahl Werte in der Spalte G
    i = 4
    Do
        If Cells(i, 3) > 0 Then j = i
        i = i + 1
    Loop While Cells(i, 3) <> 0
    ,
'Tranponiert die Werte aus G nach I
'im Anschluss an die vorhandenen Daten
    For i = 4 To j
        k = k + 1
        Cells(k, 9) = Cells(i, 3)
    Next i
End Sub
```

```
Option Explicit
Sub Tabelle_leer()
    Worksheets("Zufallszahlen").Cells.Clear
End Sub
```

```
Sub Zufallszahlen()
    Dim i, j, n As Long
    Dim k, m As Integer
    Dim dx, x As Double
    Worksheets("Zufallszahlen").Activate
    n = InputBox("Bitte eingeben..", "Anzahl Zufallszahlen", 1000)
    k = Int(1 + 3.3 * Log(n) / Log(10#))
    ReDim T(k)
    For i = 1 To k
        T(i) = 0
    Next i
    dx = 1 / k
    x = 0
    Randomize
    For i = 1 To n
        x = Rnd(x)
        j = 1: m = 0
        Do
            If x < j / k Then
                T(j) = T(j) + 1
                Cells(T(j), j) = x
                m = 1
            End If
            j = j + 1
        Loop While m = 0
    Next i
End Sub
```

```
Sub Grafik_einblenden()
    Dim i, j, k, a() As Long
    Dim b, h, n, max As Double
    Worksheets("Zufallszahlen").Activate
    b = 60
    n = 0
    max = 0
    Do
        n = n + 1
    Loop While Cells(1, n) <> ""
    n = n - 1
    ReDim a(n)
    For i = 1 To n
        j = 0
        Do
            j = j + 1
            If Cells(j, i) <> "" Then
                k = j
            End If
        Loop While Cells(j, i) <> ""
        a(i) = k
        If k > max Then max = k
    Next i
    For i = 1 To n
        h = Int(a(i) / max * 500)
        'Parameter: Type, Left, Top, Width, Height
        Shapes.AddShape msoShapeRectangle, (i - 1) * b, 0, b, h
    Next i
End Sub
```

```
Sub Grafik_ausblenden()
    Dim Shp As Shape
    Worksheets("Zufallszahlen").Activate
    'alle Charts löschen
    For Each Shp In Shapes
        Shp.Delete
    Next
End Sub
```

```

Sub Zufallszahlen_II()
Dim i, j, n, max, min As Long
Dim k, m As Integer
Dim dx, x, b, h, dif, pdif, d, p As Double
Worksheets("Zufallszahlen").Activate
n = InputBox("Bitte eingeben..", "Anzahl Zufallszahlen", 1000)
k = Int(1 + 3.3 * Log(n) / Log(10#))
ReDim T(k) As Long
For i = 1 To k
    T(i) = 0
Next i
dx = 1 / k
x = 0
Randomize
For i = 1 To n
    x = Rnd(x)
    m = 0
    j = 1
    Do
        If x < j / k Then
            T(j) = T(j) + 1
            m = 1
        End If
        j = j + 1
    Loop While m = 0
    max = j - 1
Next i
max = 0: min = 0
For i = 1 To k
    If i = 1 Then min = T(i)
    If T(i) < min Then min = T(i)
    If T(i) > max Then max = T(i)
Next i
b = 60
For i = 1 To k
    h = Int(T(i) / max * 400)
    'Parameter: Type, Left, Top, Width, Height
    Shapes.AddShape msoShapeRectangle, (i - 1) * b, 51, b, h

    d = max - min
    p = 100 - Int(T(i) / max * 1000) / 10
    If i = 1 Then
        dif = d
        pdif = p
    Else
        If d > dif Then dif = d
        If p > pdif Then pdif = p
    End If
    Cells(1, i) = T(i)
    Cells(2, i) = max - T(i)
    Cells(3, i) = Int(T(i) / max * 1000) / 10
Next i
Cells(1, k + 1) = max
Cells(2, k + 1) = dif
Cells(3, k + 1) = pdif
End Sub

```

```
Sub Normalverteilung()  
    Dim x, y, s As Double  
    Dim i, n As Long  
    Dim j As Integer  
    Worksheets("Zufallszahlen").Activate  
    n = InputBox("Bitte eingeben..", "Anzahl Zufallszahlen", 1000)  
    x = 0  
    Randomize  
    For i = 1 To n  
        s = 0  
        For j = 1 To 12  
            x = Rnd(x)  
            s = s + x  
        Next j  
        y = s - 6  
        Cells(i, 1) = y  
    Next i  
    Columns("A:A").Select  
    Selection.Sort Key1:=Range("A1"), _  
        Order1:=xlAscending, _  
        Header:=xlGuess, _  
        OrderCustom:=1, MatchCase:=False, _  
        Orientation:=xlTopToBottom, _  
        DataOption1:=xlSortNormal  
End Sub  
Sub Normalverteilungs_Diagramm()  
    Charts.Add  
    ActiveChart.ChartType = xlLine  
    ActiveChart.SetSourceData  
        Source:=Sheets("Zufallszahlen").Range("A1:A1000"), PlotBy:=xlColumns  
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Zufallszahlen"  
    With ActiveChart  
        .HasTitle = True  
        .ChartTitle.Characters.Text = "Normalverteilung"  
        .Axes(xlCategory, xlPrimary).HasTitle = True  
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "x"  
        .Axes(xlValue, xlPrimary).HasTitle = True  
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "y"  
    End With  
    ActiveChart.HasLegend = False  
    ActiveChart.HasDataTable = False  
End Sub
```

```
Option Explicit
```

```
Sub Normalverteilung_Formblatt()  
Worksheets("Fertigung").Activate  
Worksheets("Fertigung").Cells.Clear  
Range("A1") = "Mittelwert"  
Range("A2") = "Standardabweichung"  
Range("A3") = "Toleranzbereich +/-"  
Range("A4") = "Anzahl Zufallszahlen"  
Range("A6") = "Ausschuss - Anzahl"  
Range("A7") = "          - %" "  
Range("A:A").ColumnWidth = 20  
Range("B:B").ColumnWidth = 10  
End Sub
```

```
Sub Normalverteilung()  
Dim x, y, su, mi, ab, tb As Double  
Dim i, n, m As Long  
Dim j, k As Integer  
DoEvents  
Worksheets("Fertigung").Activate  
mi = Cells(1, 2)  
ab = Cells(2, 2)  
tb = Cells(3, 2)  
n = Cells(4, 2)  
x = 0  
Randomize  
For k = 1 To 10  
    m = 0  
    For i = 1 To n  
        su = 0  
        For j = 1 To 12  
            x = Rnd(x)  
            su = su + x  
        Next j  
        y = ab * (su - 6) + mi  
        If Abs(mi - y) > tb Then m = m + 1  
    Next i  
    Cells(6, 1 + k) = m  
    Cells(7, 1 + k) = Int(1000 / n * m) / 10  
Next k  
End Sub
```

```

Option Explicit
Public Psl As Double
Public objDiagramm As Object 'As Worksheet
Public DTitel, xTitel, yTitel As String
Public intLeft, intTop, intWidth, intHeight As Integer
Sub Ausgabe_Formblatt()
    Dim MyDoc As Object
    Dim Shp As Shape
    Set MyDoc = ThisWorkbook.Worksheets("Ausgabe")
    MyDoc.Activate
    MyDoc.Cells.Clear
,
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
'Neue Beschriftung
    Range("A1") = "Anzahl Bedienpersonal"
    Range("A2") = "Anzahl Ausleiher/Minute"
    Range("A3") = "Bedienzeit in Sekunden"
    Range("A4") = "Wahrscheinlichkeit (<=1)"

    Range("A6") = "Laufzeit [Min.]"
    Range("A7") = "Zufallszahl"
    Range("A8") = "Istzeit [Sek.]"
    Range("A9") = "Ausleiher"
    Range("B9") = "Gesamt"
    Range("C9") = "Schnitt"
    Range("D9") = "maxWart"
    Range("E9") = "SchWart"
    Range("A14") = "Bedienung"
    Range("B14") = "aktuell [s]"
    Range("C14") = "Summe [s]"
    Range("D14") = "Schnitt"
    Columns("A:E").EntireColumn.AutoFit
End Sub
Sub Ausgabe_Testdaten()
    Dim i As Integer
    Cells(1, 2) = 4
    Cells(2, 2) = 11
    Cells(6, 2) = 60
    Cells(3, 2) = 20
    Cells(4, 2) = 0.1
    Cells(3, 3) = 30
    Cells(4, 3) = 0.1
    Cells(3, 4) = 40
    Cells(4, 4) = 0.1
    Cells(3, 5) = 50
    Cells(4, 5) = 0.2
    Cells(3, 6) = 60
    Cells(4, 6) = 0.2
    Cells(3, 7) = 70
    Cells(4, 7) = 0.2
    Cells(3, 8) = 80
    Cells(4, 8) = 0.1
End Sub
Sub Ausgabe_Simulation()
    Dim i, j, k, Al, Ab, Bd(), Aw, MaxAl, SumAl, SumWa As Long
    Dim x, y, Tb(), ew As Double
    Dim Zl As String
'Initialisierung
    Ab = Cells(1, 2) 'Anzahl Bedienung
    ReDim Bd(Ab, 3) 'Bedienzeiten
    For i = 1 To Ab
        For j = 1 To 3
            Bd(i, j) = 0
        Next j
        Cells(14 + i, 1) = i
    Next i
'Anzahl Wahrscheinlichkeiten
'muss in der Summe 1 ergeben!
    Aw = 0
    Do
        Aw = Aw + 1
    Loop While Cells(4, 1 + Aw) > 0
    Aw = Aw - 1 'Anzahl Werte
    ReDim Tb(Aw, 2)

```

```

'Wahrscheinlichkeiten in Vektor speichern
ew = 0
For i = 1 To Aw
    Tb(i, 1) = Cells(3, 1 + i)
    ew = ew + Cells(4, 1 + i)
    Tb(i, 2) = ew
Next i
'Initialisierung
Al = 0
MaxAl = 0
DoEvents
i = 0
Randomize (2)
'Start
Do
'Zeit
    i = i + 1
    Cells(5, 2) = i
'vorhandene Bedienzeiten um eine Sekunde reduzieren und anzeigen
    For j = 1 To Ab
        If Bd(j, 1) > 0 Then
            Bd(j, 1) = Bd(j, 1) - 1
            Bd(j, 3) = Bd(j, 3) + 1
        End If
        Cells(14 + j, 2) = Bd(j, 1)
        Cells(14 + j, 3) = Bd(j, 2)
    Next j
'Zufallszahl 1 erzeugen und anzeigen
    x = Rnd(x)
    Cells(7, 2) = Round(x, 3)
'Wahrscheinlichkeit des Erscheinens eines Ausleihers (Anzahl/60s)
    If x <= (Cells(2, 2) / 60) Then
        'Ausleiher da
        Al = Al + 1
        SumAl = SumAl + 1
    End If
    Cells(10, 1) = Al
'vorhandene Ausleiher zuteilen
    If Al > 0 Then
        For j = 1 To Ab
            If Bd(j, 1) = 0 Then
                'Bedienzeit nach Wahrscheinlichkeit ermitteln
                '(Zufallszahl 2)
                x = Rnd(x)
                For k = 1 To Aw
                    If x >= Tb(k, 2) Then
                        y = Tb(k, 1)
                        k = Aw
                    End If
                Next k
                'Ausleiher mit Bedienzeit zuweisen
                Bd(j, 1) = y
                Bd(j, 2) = Bd(j, 2) + y 'Summenbildung
                Al = Al - 1
                j = Ab '(Ausstieg aus der Schleife)
            End If
        Next j
    End If
    Cells(10, 1) = Al
'Summe wartende Ausleiher
    Cells(i, 9) = Al
    If Al > MaxAl Then
        MaxAl = Al
        Cells(10, 4) = MaxAl
    End If
    SumWa = SumWa + Al
    Loop While i < Cells(6, 2) * 60
'Ausleiher
    Cells(10, 2) = SumAl
    x = SumAl / (Cells(6, 2) * 60)
    Cells(10, 3) = Round(x, 3)
    x = SumWa / (Cells(6, 2) * 60)
    Cells(10, 5) = Round(x, 3)
'Schnitt
    For i = 1 To Ab
        x = Bd(i, 3) / (Cells(6, 2) * 60)
        Cells(14 + i, 4) = Round(x, 3)
    
```

```
Next i
Call Ausgabe_Grafik
End Sub
Sub Ausgabe_Grafik()
Columns("I:I").Select
Charts.Add
ActiveChart.ChartType = xlArea
ActiveChart.SetSourceData Source:=Worksheets("Ausgabe").Range("I1:I" +
LTrim(Str(UsedRange.Rows.Count))), PlotBy _
:=xlColumns
ActiveChart.Location Where:=xlLocationAsObject, Name:="Ausgabe"
With ActiveChart
.HasTitle = True
.ChartTitle.Characters.Text = "Simulation Werkzeugausgabe"
.Axes(xlCategory, xlPrimary).HasTitle = True
.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Zeit"
.Axes(xlValue, xlPrimary).HasTitle = True
.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Warteschlange"
End With
End Sub
```

```
Option Explicit
Global Maschinenanzahl, Produktmenge As Integer
Sub Belegung_Neu()
    Dim i, x, y As Integer
    Dim Z1, Sp As String
    y = Val(frmBelegung.TextBox1)
    x = Val(frmBelegung.TextBox2)
    Maschinenanzahl = y
    Produktmenge = x
    Worksheets("Belegung").Activate
    Worksheets("Belegung").Cells.Clear
    For i = 1 To y
        Sp = Chr$(66 + i)
        Range(Sp + "02").Value = "M" + Right("00" + LTrim(Str(i)), 2)
        Columns(Sp).EntireColumn.AutoFit
    Next i
    Sp = Chr$(66 + y + 1)
    Range(Sp + "02").Value = "FARBE"
    Columns(Sp).EntireColumn.AutoFit
    Sp = Chr$(66 + y + 4)
    Range(Sp + "02").Value = "FOLGE"
    Columns(Sp).EntireColumn.AutoFit
    For i = 1 To x
        Z1 = Right("00" + LTrim(Str(i + 2)), 2)
        Range("B" + Z1).Value = "P" + Right("00" + LTrim(Str(i)), 2)
    Next i
    Columns("B").EntireColumn.AutoFit
    Range("C03").Activate
End Sub
```

```
Sub Belegung_Testdaten()
    frmBelegung.TextBox1 = 3
    frmBelegung.TextBox2 = 5
    Call Belegung_Neu
    Range("C03").Value = "5"
    Range("C04").Value = "5"
    Range("C05").Value = "12"
    Range("C06").Value = "4"
    Range("C07").Value = "8"

    Range("D03").Value = "3"
    Range("D04").Value = "3"
    Range("D05").Value = "4"
    Range("D06").Value = "2"
    Range("D07").Value = "2"

    Range("E03").Value = "8"
    Range("E04").Value = "4"
    Range("E05").Value = "2"
    Range("E06").Value = "7"
    Range("E07").Value = "9"
    Call Belegung_Johnson
End Sub
```

```
Sub Belegung_Johnson()
    Dim i, i1, i2, j, k, pr, pc, c, n As Integer
    Dim mk1, mk2, Min, q, dz1, dz2, Esp, x As Long
    Dim l1, t1, w1, h1 As Integer
    Dim Z1, Sp, Sp1, Sp2, C1 As String
    Dim MyDocument As Worksheet
    j = Maschinenanzahl
    k = Produktmenge
    ReDim M1(k), M2(k), mk(k)

'Initialisierung
'und Bestimmung der Reihenfolge nach Johnson
    For i = 1 To k
        mk(i) = i
    Next i
    mk1 = 0
    mk2 = 0
    For i1 = 1 To k
        pr = 0
        Min = 0
        Do
            pr = pr + 1
            If mk(pr) > 0 Then
```

```

    Z1 = Right("00" + LTrim(Str(pr + 2)), 2)
    Min = Val(Range("C" + Z1).Value)
    pc = 1
End If
Loop While Min = 0
For i2 = 1 To k
    If mk(i2) > 0 Then
        Z1 = Right("00" + LTrim(Str(i2 + 2)), 2)
        C1 = Val(Range("C" + Z1).Value)
        If Val(C1) < Min Then
            Min = Val(C1)
            pc = 1
            pr = i2
        End If
        Sp = Chr$(66 + j)
        Z1 = Right("00" + LTrim(Str(i2 + 2)), 2)
        C1 = Val(Range(Sp + Z1).Value)
        If Val(C1) < Min Then
            Min = Val(C1)
            pc = j
            pr = i2
        End If
    End If
End If
Next i2
If pc = 1 Then
    mk1 = mk1 + 1
    M1(mk1) = pr
Else
    mk2 = mk2 + 1
    M2(mk2) = pr
End If
mk(pr) = 0
Next i1

```

'Reihenfolgeausgabe

```

Sp = Chr$(66 + j + 2)
Range(Sp + "02").Value = "Gesetzte Reihenfolge"
Columns(Sp).EntireColumn.AutoFit
ReDim Su(j)
For i = 1 To j
    Su(i) = 0
Next i
For i1 = 1 To k
    Z1 = Right("00" + LTrim(Str(i1 + 2)), 2)
    x = Val(Range("C" + Z1).Value)
    Su(1) = Su(1) + x
    For i2 = 2 To j
        Sp = Chr$(66 + i2)
        Z1 = Right("00" + LTrim(Str(i1 + 2)), 2)
        x = Val(Range(Sp + Z1).Value)
        If Su(i2 - 1) > Su(i2) Then
            Su(i2) = Su(i2 - 1) + x
        Else
            Su(i2) = Su(i2) + x
        End If
    Next i2
    Sp = Chr$(66 + j + 2)
    Z1 = Right("00" + LTrim(Str(i1 + 2)), 2)
    Range(Sp + Z1).Value = Str(Su(j))
Next i1
Sp = Chr$(66 + j + 2)
Z1 = Right("00" + LTrim(Str(k + 3)), 2)
Range(Sp + Z1).Value = "Durchlaufzeit:" + Str$(Su(j))
dz1 = Su(j)
If dz1 > 0 Then
    q = Int(10000 / dz1)
Else
    q = 1
End If

```

'Lösungsausgabe

```

Sp = Chr(66 + j + 3)
Range(Sp + "02").Value = "Optimierte Reihenfolge"
For i = 1 To j
    Sp1 = Chr(66 + j + 3 + i * 2)
    Sp2 = Chr(66 + j + 4 + i * 2)
    Range(Sp1 + "02:" + Sp2 + "02").MergeCells = True
    Range(Sp1 + "02").Value = "M" + Right("00" + LTrim(Str(i)), 2)

```

```

Range(Sp2 + "02").Value = "M" + Right("00" + LTrim(Str(i)), 2)
Next i
Columns(Sp).EntireColumn.AutoFit
ReDim mg(j)
For i = 1 To j
    mg(i) = 0
Next i
If mk1 > 0 Then
    For i = 1 To mk1
        n = Val(M1(i))
        Sp = Chr$(66 + j + 4)
        Z1 = Right("00" + LTrim(Str(n + 2)), 2)
        Range(Sp + Z1).Value = Str(i)
        x = Val(Range("C" + Z1).Value)
        c = c + 1
        'von
        Sp = Chr$(66 + j + 5)
        Range(Sp + Z1).Value = mg(1)
        'bis
        Sp = Chr$(66 + j + 6)
        Range(Sp + Z1).Value = mg(1) + x
        mg(1) = mg(1) + x
        For i2 = 2 To j
            Sp = Chr$(66 + i2)
            Z1 = Right("00" + LTrim(Str(n + 2)), 2)
            x = Val(Range(Sp + Z1).Value)
            If mg(i2 - 1) > mg(i2) Then
                'von
                Sp = Chr$(66 + j + 3 + i2 * 2)
                Range(Sp + Z1).Value = mg(i2 - 1)
                'bis
                Sp = Chr$(66 + j + 4 + i2 * 2)
                Range(Sp + Z1).Value = mg(i2 - 1) + x
                mg(i2) = mg(i2 - 1) + x
            Else
                'von
                Sp = Chr$(66 + j + 3 + i2 * 2)
                Range(Sp + Z1).Value = mg(i2)
                'bis
                Sp = Chr$(66 + j + 4 + i2 * 2)
                Range(Sp + Z1).Value = mg(i2) + x
                mg(i2) = mg(i2) + x
            End If
        Next i2
        Sp = Chr$(66 + j + 3)
        Z1 = Right("00" + LTrim(Str(M1(i) + 2)), 2)
        Range(Sp + Z1).Value = Str(mg(j))
    Next i
End If
If mk2 > 0 Then
    For i = mk2 To 1 Step -1
        n = Val(M2(i))
        Sp = Chr$(66 + j + 4)
        Z1 = Right("00" + LTrim(Str(n + 2)), 2)
        Range(Sp + Z1).Value = Str(mk1 + i)
        x = Val(Range("C" + Z1).Value)
        'von
        Sp = Chr$(66 + j + 5)
        Range(Sp + Z1).Value = mg(1)
        'bis
        Sp = Chr$(66 + j + 6)
        Range(Sp + Z1).Value = mg(1) + x
        mg(1) = mg(1) + x
        For i2 = 2 To j
            Sp = Chr$(66 + i2)
            Z1 = Right("00" + LTrim(Str(n + 2)), 2)
            x = Val(Range(Sp + Z1).Value)
            If mg(i2 - 1) > mg(i2) Then
                'von
                Sp = Chr$(66 + j + 3 + i2 * 2)
                Range(Sp + Z1).Value = mg(i2 - 1)
                'bis
                Sp = Chr$(66 + j + 4 + i2 * 2)
                Range(Sp + Z1).Value = mg(i2 - 1) + x
                mg(i2) = mg(i2 - 1) + x
            Else
                'von

```

```

        Sp = Chr$(66 + j + 3 + i2 * 2)
        Range(Sp + Z1).Value = mg(i2)
        'bis
        Sp = Chr$(66 + j + 4 + i2 * 2)
        Range(Sp + Z1).Value = mg(i2) + x
        mg(i2) = mg(i2) + x
    End If
Next i2
Sp = Chr$(66 + j + 3)
Z1 = Right("00" + LTrim(Str(M2(i) + 2)), 2)
Range(Sp + Z1).Value = Str(mg(j))
Next i
End If
Sp = Chr$(66 + j + 3)
Z1 = Right("00" + LTrim(Str(k + 3)), 2)
Range(Sp + Z1).Value = "Durchlaufzeit:" + Str$(mg(j))
dz2 = mg(j)
If dz1 > 0 Then
    Esp = 100 / dz1 * (dz1 - dz2)
Else
    Esp = 0
End If
Sp = Chr$(66 + j + 3)
Z1 = Right("00" + LTrim(Str(k + 4)), 2)
Range(Sp + Z1).Value = "Einsparung: " + Format(Esp, "####.00") + " %"
End Sub
Sub Belegung_Grafik()
    Dim j, k, l1, t1, w1, h1, i, i1, i2 As Integer
    Dim Z1, Sp, Spf, cl1, cl2 As String
    Dim FarbNr As Long
    Dim ZRot, ZGruen, ZBlau As Byte
    Dim fc(15) As Long
    Dim MyDoc As Worksheet
    Set MyDoc = Worksheets("Belegung")
    j = Maschinenanzahl
    k = Produktmenge
    If j = 0 Or k = 0 Then
        MsgBox "Keine Daten vorhanden!", vbCritical + vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    'den Produkten Farben zuordnen
    fc(0) = RGB(0, 0, 0)           'Schwarz
    fc(1) = RGB(0, 0, 128)        'Blau
    fc(2) = RGB(0, 128, 0)        'Grün
    fc(3) = RGB(64, 128, 128)    'Cyan
    fc(4) = RGB(128, 0, 0)        'Rot
    fc(5) = RGB(128, 0, 128)     'Magenta
    fc(6) = RGB(255, 255, 0)     'Gelb
    fc(7) = RGB(222, 222, 222)   'Weiß
    fc(8) = RGB(128, 128, 128)   'Grau
    fc(9) = RGB(0, 0, 255)       'Hellblau
    fc(10) = RGB(0, 255, 0)      'Hellgrün
    fc(11) = RGB(0, 255, 255)    'Hellcyan
    fc(12) = RGB(255, 0, 0)      'Hellrot
    fc(13) = RGB(255, 0, 255)    'Hellmagenta
    fc(14) = RGB(255, 255, 128)  'Hellgelb
    fc(15) = RGB(255, 255, 255)  'Hellweiß
    Spf = Chr$(66 + j + 1)
    For i2 = 1 To k
        Z1 = Right("00" + LTrim(Str(i2 + 2)), 2)
        Range(Spf + Z1).Interior.Color = QBColor(i2 Mod 16)
    Next i2
    'Anzeige
    For i1 = 1 To j
        For i2 = 1 To k
            Z1 = Right("00" + LTrim(Str(i2 + 2)), 2)
            Sp = Chr$(66 + j + 3 + i1 * 2)
            cl1 = Range(Sp + Z1).Value
            Sp = Chr$(66 + j + 4 + i1 * 2)
            cl2 = Range(Sp + Z1).Value

            l1 = 10 * Val(cl1)
            t1 = 200 + (i1 - 1) * 10
            w1 = 10 * (Val(cl2) - Val(cl1))
            h1 = 10

```

```
MyDoc.Shapes.AddShape msoShapeRectangle, l1, t1, w1, h1
i = MyDoc.Shapes.Count
```

```
'RGB Farbe bestimmen
```

```
Z1 = Right("00" + LTrim(Str(i2 + 2)), 2)
FarbNr = Range(Spf + Z1).Interior.Color
If Len(Hex(FarbNr)) > 0 Then _
    ZRot = CByte("&h" & Right(Hex(FarbNr), 2))
If Len(Hex(FarbNr)) > 2 Then _
    ZGruen = CByte("&h" & Mid(Hex(FarbNr), 3, 2))
If Len(Hex(FarbNr)) > 4 Then _
    ZBlau = CByte("&h" & Mid(Hex(FarbNr), 5, 2))
MyDoc.Shapes(i).Fill.ForeColor.RGB = fc(i2)
```

```
Next i2
```

```
Next i1
```

```
End Sub
```

```
Sub Belegung_Grafik_Löschen()
```

```
Dim j, k, i As Integer
Dim Sp, Z1 As String
Dim MyDoc As Worksheet
Set MyDoc = Worksheets("Belegung")
j = Maschinenanzahl
k = Produktmenge
```

```
'alle Shapes löschen
```

```
Dim o As Object
For Each o In MyDoc.Shapes
    o.Delete
Next
```

```
'Farbzuordnung löschen
```

```
Sp = Chr$(66 + j + 1)
For i = 1 To k
    Z1 = Right("00" + LTrim(Str(i + 2)), 2)
    Range(Sp + Z1).Cells.Interior.ColorIndex = xlNone
Next i
```

```
End Sub
```

```
Option Explicit
Sub comStart_Click()
    Call Belegung_Formblatt
End Sub
Sub Belegung_Formblatt()
    Dim i, x, y As Integer
    Dim Z1, Sp As String
    y = Val(frmBelegung.TextBox1)
    x = Val(frmBelegung.TextBox2)
    Maschinenanzahl = y
    Produktmenge = x
    Worksheets("Belegung").Activate
    Worksheets("Belegung").Cells.Clear
    For i = 1 To y
        Sp = Chr$(66 + i)
        Range(Sp + "02").Value = "M" + Right("00" + LTrim(Str(i)), 2)
        Columns(Sp).EntireColumn.AutoFit
    Next i
    Sp = Chr$(66 + y + 1)
    Range(Sp + "02").Value = "FARBE"
    Columns(Sp).EntireColumn.AutoFit
    Sp = Chr$(66 + y + 4)
    Range(Sp + "02").Value = "FOLGE"
    Columns(Sp).EntireColumn.AutoFit
    For i = 1 To x
        Z1 = Right("00" + LTrim(Str(i + 2)), 2)
        Range("B" + Z1).Value = "P" + Right("00" + LTrim(Str(i)), 2)
    Next i
    Columns("B").EntireColumn.AutoFit
    Range("C03").Activate
    Unloadme
End Sub
```

```
Option Explicit
Sub Losgroesse_Eingabe()
    Load frmLosgroesse
    frmLosgroesse.Show
End Sub
Sub Losgroesse_Diagramm()
    Call Losgroesse_Datengrafik
End Sub
Sub Losgroesse_Diagramm_Loeschen()
    Dim MyDoc As Worksheet
    Set MyDoc = ThisWorkbook.Worksheets("Losgroesse")
    Dim Shp As Shape
    ,
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
End Sub
```

```

Private Sub cmdTest_Click()
    TextBox1 = "450"
    TextBox2 = "22"
    TextBox3 = "1000"
    TextBox4 = "12"
    TextBox5 = "15"
End Sub

Private Sub cmdStart_Click()
    Dim Kr, Kh, xj, Bz, Lz As Double
    Dim x, Kgr, Kgl, Kg, Kgs, KMin As Double
    Dim i, iMin As Integer

    Worksheets("Losgroesse").Activate
    Worksheets("Losgroesse").Cells.Clear
    Range("A1").Value = "Losgröße" & vbLf & "[Stück]"
    Range("B1").Value = "Gesamt-" & vbLf & "Rüstkosten" & vbLf & "[Euro]"
    Range("C1").Value = "Gesamt-" & vbLf & "Lagerkosten" & vbLf & "[Euro]"
    Range("D1").Value = "Gesamt-" & vbLf & "Kosten" & vbLf & "[Euro]"
    Range("E1").Value = "Gesamt-" & vbLf & "Kosten/Stück" & vbLf & "[Euro]"
    Columns("A:E").EntireColumn.AutoFit
    ActiveWindow.Visible = False
    Windows("BfI.xls").Activate
    Columns("B:E").Select
    Selection.NumberFormat = "0.00"
    Kr = Val(TextBox1)
    Kh = Val(TextBox2)
    xj = Val(TextBox3)
    Bz = Val(TextBox4)
    Lz = Val(TextBox5)
    i = 1
    iMin = 0
    KMin = 0
    For x = xj / 100 To xj Step (xj / 100)
        Kgr = xj / x * Kr           'Gesamtrüstkosten
        Kgl = x / 2 * Kh * (Bz + Lz) / 100 'Gesamtlagerkosten
        Kg = Kgr + Kgl             'Gesamtkosten
        Kgs = Kg / xj             'Gesamtkosten/Stück
        i = i + 1
    Next x
    'Minimum bestimmen
    If iMin = 0 Then
        KMin = Kg
        iMin = i
    Else
        If Kg < KMin Then
            KMin = Kg
            iMin = i
        End If
    End If
    'Eintrag in Tabelle
    Z1 = Right("000" + LTrim(Str(i)), 3)
    Range("A" + Z1).Value = Round(x, 2)
    Range("B" + Z1).Value = Round(Kgr, 2)
    Range("C" + Z1).Value = Round(Kgl, 2)
    Range("D" + Z1).Value = Round(Kg, 2)
    Range("E" + Z1).Value = Round(Kgs, 2)
    Next x
    Z1 = Right("000" + LTrim(Str(iMin)), 3)
    Range("A" + Z1 + ":E" + Z1).Interior.Color = vbYellow
    Range("A" + Z1 + ":E" + Z1).Select
    Unload Me
End Sub

```

```
Option Explicit
Public MyDoc As Object 'As Worksheet
Public DTitel, xTitel, yTitel As String
Sub Losgroesse_Datengrafik()
    Dim lngNumRows, lngNumCols As Long
'Verweis auf Worksheet mit Daten
    Set MyDoc = ThisWorkbook.Worksheets("Losgroesse")
'Übergabe der Anzahl der Spalten/Zeilen:
    lngNumRows = MyDoc.UsedRange.Rows.Count
    lngNumCols = MyDoc.UsedRange.Columns.Count
'Neues Diagramm
    Charts.Add
    ActiveChart.ChartType = xlLineStacked100
    ActiveChart.SetSourceData Source:=Sheets("Losgroesse"). _
        Range("A1:E" + LTrim(Str(lngNumRows))), PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Losgroesse"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "KOSTENVERLAUF"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Losgröße"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Kosten"
    End With
End Sub
```

```

Option Explicit
Sub Rotation_Formblatt()
    Dim MyDoc As Object
    Dim Shp As Shape
    Set MyDoc = ThisWorkbook.Worksheets("Rotation")
    MyDoc.Activate
    MyDoc.Cells.Clear
,
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
'Neue Beschriftung
    Range("A1") = "Gefäßradius r [m]"
    Range("A2") = "Spiegelhöhe y0 [m]"
    Range("A3") = "Winkelgeschwindigkeit w [1/s]"
    Columns("A:E").EntireColumn.AutoFit
End Sub
Sub Rotation_Auswertung()
    Dim x, y, r, y0, w, g, c As Double
    Dim i As Integer
    g = 9.81 'Erdbeschleunigung [m/s^2]
    r = Cells(1, 2)
    y0 = Cells(2, 2)
    w = Cells(3, 2)
    c = y0 - (w * w * r * r / (4 * g))
    i = 0
    For x = 0 To r Step r / 100
        y = w * w * x * x / (2 * g) + c
        i = i + 1
        Cells(i, 3) = y
    Next x
End Sub
Sub Rotation_zeigen()
    Charts.Add
    ActiveChart.ChartType = xlLine
    ActiveChart.SetSourceData Source:=Sheets("Rotation").Range("C1:C100"), _
        PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).Name = ""Flüssigkeitsspiegel""
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Rotation"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Rotationsparaboloid"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "x"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "y"
    End With
    ActiveWindow.Visible = False
End Sub

```

Option Explicit

Sub Strömung_Formblatt()

```
Dim MyDoc As Object
Dim Shp As Shape
Set MyDoc = ThisWorkbook.Worksheets("Strömung")
MyDoc.Activate
MyDoc.Cells.Clear
```

'alle Charts löschen

```
For Each Shp In MyDoc.Shapes
    Shp.Delete
```

Next

'Neue Beschriftung

```
Range("A1") = "Rohrlänge l [m]"
Range("A2") = "Rohrinnenradius r0 [m]"
Range("A3") = "Druckdifferenz p [N/cm" + Chr(178) + "]"
Range("A4") = "Flüssigkeitsdichte [kg/dm" + Chr(179) + "]"
Range("A5") = "Dynamische Zähigkeit [Ns/m" + Chr(178) + "]"
Range("A6") = "Neigungswinkel [Grad]"
Range("A8") = "Mittl.Strömungsgeschw. vm [m/s]"
Range("A9") = "Volumenstrom Q [m" + Chr(179) + "/s]"
Columns("A:E").EntireColumn.AutoFit
```

End Sub

Sub Strömung_Auswertung()

```
Dim r, r0, l, p, d, u, a, sa, g, v, vm, Q As Double
Dim i As Integer
```

```
l = Cells(1, 2)
r0 = Cells(2, 2)
p = Cells(3, 2)
d = Cells(4, 2)
u = Cells(5, 2)
a = Cells(6, 2)
sa = Sin(a * 3.1415926 / 180)
For r = 0 To r0 Step r0 / 100
    v = (p * 10000 + l * d * 1000 * g * sa) / (4 * l * u) * (r0 * r0 - r * r)
    i = i + 1
    Cells(i, 3) = r
    Cells(i, 4) = v
Next r
vm = r0 * r0 * (p * 10000 + l * d * 1000 * g * sa) / (8 * l * u)
Q = vm * r0 * r0 * 3.1415926
Cells(8, 2) = vm
Cells(9, 2) = Q
```

End Sub

Sub Strömung_zeigen()

```
Columns("D:D").Select
Charts.Add
ActiveChart.ChartType = xlLine
ActiveChart.SetSourceData Source:=Sheets("Strömung").Range("D1:D100"), _
    PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = ""Strömungsgeschwindigkeit""
ActiveChart.Location Where:=xlLocationAsObject, Name:="Strömung"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Strömungsgeschwindigkeit"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "r [m]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "v [m/s]"
End With
ActiveChart.HasDataTable = False
ActiveChart.Legend.Select
Selection.Delete
```

End Sub

```

Option Explicit
Sub Wandtemperatur_Formblatt()
  Dim MyDoc As Object
  Dim Shp As Shape
  Set MyDoc = ThisWorkbook.Worksheets("Wandtemperatur")
  MyDoc.Activate
  MyDoc.Cells.Clear
,
'alle Charts löschen
  For Each Shp In MyDoc.Shapes
    Shp.Delete
  Next
'Neue Beschriftung
  Range("A1") = "Wandstärke x [m]"
  Range("A2") = "Schrittzahl n (max=95)"
  Range("A3") = "Wärmeleitfähigkeit L [W/m grd]"
  Range("A4") = "Spez. Wärmekapazität c [kJ/kg grd]"
  Range("A5") = "Dichte d [kg/" + Chr(179) + "m]"
  Range("A6") = "Wärmeübergangszahl-Innen a1 [W/m" + Chr(178) + " grd]"
  Range("A7") = "Wärmeübergangszahl-Aussen a2 [W/m" + Chr(178) + " grd]"
  Columns("A:B").EntireColumn.AutoFit
End Sub

Sub Wandtemperatur_Testdaten()
  Cells(1, 2) = 1
  Cells(2, 2) = 10
  Cells(3, 2) = 1.2
  Cells(4, 2) = 1
  Cells(5, 2) = 2000
  Cells(6, 2) = 8
  Cells(7, 2) = 5
End Sub

Sub Wandtemperatur_Verlauf()
  Dim t(99), tx(99) As Double
  Dim x, L, c, d, Ti, Ta, T7, Tmin, Tmax As Double
  Dim a, a1, a2, dx, dt, tt, xi, g As Double
  Dim n, i, j, an As Integer
  tt = 0
,
'Daten
  x = Cells(1, 2)
  n = Cells(2, 2)
  L = Cells(3, 2)
  c = Cells(4, 2)
  d = Cells(5, 2)
  Ti = Cells(8, 2)
  Ta = Cells(9, 2)
,
'Bestimmung der Schrittweiten
  If n = 0 Then
    dx = 0
  Else
    dx = x / n
  End If
  a = L / (c * d)
  If a = 0 Then
    dt = 0
  Else
    dt = (dx * dx) / (2 * a)
  End If
,
'Temperaturverteilung
  Range("C:" + Chr$(66 + n + 4)).ColumnWidth = 6
  Columns("C:" + Chr$(66 + n + 4)).Select
  Selection.NumberFormat = "0.0"
  Range("A13") = "Temperatur (an der Stelle x)"
  Range("B14") = "Zeit"
  Range("A14") = "Eingabe Start-Temperaturverlauf "
  Range("B14") = 0
  Cells(13, 3) = "Ti"
  Cells(13, 4) = "T(0)"
  For i = 1 To n
    xi = (i - 0.5) * dx
    Cells(13, 4 + i) = "T(" + LTrim(Str(xi)) + ")"
  Next i
  Cells(13, 4 + n + 1) = "T(" + LTrim(Str(x)) + ")"

```

```

Cells(13, 4 + n + 2) = "Ta"
Range("C14").Select
Range("A10") = "Anzahl der Durchläufe"
End Sub

Sub Wandtemperatur_Auswertung()
Dim t(99), tx(99) As Double
Dim x, L, c, d, Ti, Ta, T7, Tmin, Tmax As Double
Dim a, a1, a2, b1, b2, dx, dt, tt, xi, g As Double
Dim n, i, j, an As Integer
tt = 0

'Daten
x = Cells(1, 2)
n = Cells(2, 2)
L = Cells(3, 2)
c = Cells(4, 2)
d = Cells(5, 2)
a1 = Cells(6, 2)
a2 = Cells(7, 2)
Ti = Cells(8, 2)
Ta = Cells(9, 2)

'Bestimmung der Schrittweiten
If n = 0 Then
dx = 0
Else
dx = x / n
End If
a = L / (c * d)
If a = 0 Then
dt = 0
Else
dt = (dx * dx) / (2 * a)
End If

'Einlesen und Hilfspunkte
Ti = Cells(14, 3)
Ta = Cells(14, 3 + n + 3)
For i = 2 To n + 1
t(i) = Cells(14, 3 + i)
Next i
b1 = (2 * L) / (a1 * dx)
b2 = (2 * L) / (a2 * dx)
t(1) = (b1 - 1) / (1 + b1) * t(2) + 2 / (1 + b1) * Ti
t(n + 2) = (b2 - 1) / (1 + b2) * t(n + 1) + 2 / (1 + b2) * Ta
Cells(14, 3 + 1) = (t(1) + t(2)) / 2
Cells(14, 3 + n + 2) = (t(n + 1) + t(n + 2)) / 2

'Berechnung der zeitlichen Veränderungen
an = Cells(10, 2)
tt = 0
For i = 1 To an
tt = tt + dt
t(1) = (b1 - 1) / (1 + b1) * t(2) + 2 / (1 + b1) * Ti
t(n + 2) = ((b2 - 1) / (1 + b2)) * t(n + 2) + (2 / (1 + b2)) * Ta
tx(1) = t(1)
tx(n + 2) = t(n + 2)
For j = 2 To n + 1
tx(j) = (t(j + 1) + t(j - 1)) / 2
Next j
For j = 1 To n + 2
t(j) = tx(j)
Next j
Cells(14 + i, 2) = Int(tt * 10) / 10
Cells(14 + i, 3) = Ti
Cells(14 + i, 3 + 1) = (t(1) + t(2)) / 2
For j = 2 To n + 1
Cells(14 + i, 3 + j) = t(j)
Next j
Cells(14 + i, 3 + n + 2) = (t(n + 1) + t(n + 2)) / 2
Cells(14 + i, 3 + n + 3) = Ta
Next i
End Sub

Sub Wandtemperatur_Zeigen()
Dim i As Integer

```

```
Charts.Add
ActiveChart.ChartType = xlLineMarkers
ActiveChart.SetSourceData Source:=Sheets("Wandtemperatur"). _
    Range("C14:P14"), PlotBy:=xlRows
For i = 1 To 19
    ActiveChart.SeriesCollection.NewSeries
Next i
For i = 1 To 19
    ActiveChart.SeriesCollection(i).Values = _
        "=Wandtemperatur!R" + LTrim(Str(14 + i)) + "C3:R" + _
            LTrim(Str(14 + i)) + "C16"
Next i
ActiveChart.Location Where:=xlLocationAsObject, Name:="Wandtemperatur"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Zeitlicher Temperaturverlauf"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "x [m]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "T [grd]"
End With
ActiveChart.Legend.Select
Selection.Delete
End Sub
```

```

Option Explicit
Public W, Wt, Qw As Double
Sub Kreisprozesse_Neustart()
    Dim MyDoc As Object
    Dim Shp As Shape
    Set MyDoc = ThisWorkbook.Worksheets("Kreisprozesse")
    MyDoc.Activate
    MyDoc.Cells.Clear
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
'Neue Beschriftung
    Range("A1") = "Masse m [kg]"
    Range("A2") = "Gaskonstante R [J/kg K]"
    Range("A3") = "Adiabatexponent x"
    Range("A4") = "Schrittanzahl"
    Range("A7") = "Zustand Nr."
    Range("A8") = "Zustandsänderung"
    Range("A9") = "p [Pa]"
    Range("A10") = "v [m" + Chr(179) + " /kg]"
    Range("A11") = "T [grd K]"
'Wertetabelle
    Range("G1") = "->"
    Range("H1") = "v [m" + Chr(179) + " /kg]"
    Range("I1") = "p [Pa]"
    Range("J1") = "W [J]"
    Range("K1") = "Wt [J]"
    Range("L1") = "Q [J]"
    Columns("G:G").ColumnWidth = 2.29
    Columns("H:L").Select
    Selection.NumberFormat = "0.000"
    Columns("A:B").EntireColumn.AutoFit
End Sub
Sub Kreisprozesse_Testdaten()
    Cells(1, 2) = 1
    Cells(2, 2) = 29.27
    Cells(3, 2) = 1.4
    Cells(4, 2) = 10

    Cells(7, 2) = 1
    Cells(8, 2) = "Is"
    Cells(9, 2) = 20
    Cells(11, 2) = 1000

    Cells(7, 3) = 2
    Cells(8, 3) = "Ad"
    Cells(9, 3) = 10

    Cells(7, 4) = 3
    Cells(8, 4) = "Is"
    Cells(9, 4) = 2

    Cells(7, 5) = 4
    Cells(8, 5) = "Ad"
    Cells(9, 5) = 4
End Sub
Sub Kreisprozesse_Auswertung()
    Dim i, j, k As Integer
    Dim n As Long
    Dim p1, p2, v1, v2, T1, T2, R, x, Q As Double
    Dim Art As String
    R = Cells(2, 2)
    x = Cells(3, 2)
    n = Cells(4, 2)
    i = 1
    n = 1
    W = 0: Wt = 0: Q = 0
    Do
        If Cells(7, 1 + i) > 0 Then
            Art = Cells(8, 1 + i)
            p1 = Cells(9, 1 + i)
            v1 = Cells(10, 1 + i)
            T1 = Cells(11, 1 + i)
            If v1 = 0 Then
                If p1 > 0 Then
                    v1 = R * T1 / p1 / 10000
                End If
            End If
        End If
    Loop

```

```

Else
    v1 = 0
End If
Cells(10, 1 + i) = v1
End If
End If
If Cells(7, 2 + i) > 0 Then
    j = i + 2
Else
    j = 2
End If
Select Case Art
Case "Is"
    p2 = Cells(9, j)
    T2 = T1
    Cells(11, j) = T2
    If p2 > 0 Then
        Q = R * T1 * Log(p1 / p2)
    Else
        Q = 0
    End If
    Cells(11, 1 + i) = Q
    Call Kreisprozesse_Isotherme(n, i, p1, v1, p2)
Case "Ad"
    p2 = Cells(9, j)
    T2 = T1 / (p1 / p2) ^ ((x - 1) / x)
    Cells(11, j) = T2
    Call Kreisprozesse_Adiabate(n, i, p1, v1, p2, x)
End Select
i = i + 1
Loop While Cells(7, 1 + i) > 0
Cells(n + 2, 10) = W
Cells(n + 2, 11) = Wt
Cells(n + 2, 12) = Qw
End Sub
Sub Kreisprozesse_Isotherme(n, i, p1, v1, p2)
Dim dp, p, px, v, vx, Wx, Wy As Double
Dim m As Double
m = Cells(1, 2)
dp = (p2 - p1) / 10
px = p1
vx = v1
For p = p1 + dp To p2 Step dp
    v = p1 * v1 / p
    n = n + 1
    Cells(n, 7) = i
    Cells(n, 8) = v
    Cells(n, 9) = p
    Wx = (v - vx) * m * (p + px) / 2
    Cells(n, 10) = Wx
    W = W + Wx
    Wy = (p - px) * m * (v + vx) / 2
    Cells(n, 11) = Wy
    Wt = Wt + Wy
    Qw = Qw + Wx
    Cells(n, 12) = Wx
    px = p
    vx = v
Next p
End Sub
Sub Kreisprozesse_Adiabate(n, i, p1, v1, p2, x)
Dim dp, p, px, v, vx, Wx, Wy As Double
Dim m As Double
m = Cells(1, 2)
dp = (p2 - p1) / 10
px = p1
vx = v1
For p = p1 + dp To p2 Step dp
    v = (p1 * v1 ^ x / p) ^ (1 / x)
    n = n + 1
    Cells(n, 7) = i
    Cells(n, 8) = v
    Cells(n, 9) = p
    Wx = (v - vx) * m * (p + px) / 2
    Cells(n, 10) = Wx
    W = W + Wx
    Wy = (p - px) * m * (v + vx) / 2

```

```
Cells(n, 11) = Wy
Wt = Wt + Wy
px = p
vx = v
Next p
End Sub
Sub Kreisprozesse_Zeichnen()
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData
    Source:=Sheets("Kreisprozesse").Range("H2:I39"), PlotBy:=xlColumns
ActiveChart.Location Where:=xlLocationAsObject, Name:="Kreisprozesse"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "p-v-Diagramm"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "v"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "p"
End With
ActiveChart.Legend.Select
Selection.Delete
End Sub
```

Option Explicit

```

Sub Gleichstrom_Formular()
  Dim MyDoc As Object
  Dim Shp As Shape
  Set MyDoc = ThisWorkbook.Worksheets("Gleichstrom")
  MyDoc.Activate
  MyDoc.Cells.Clear
,
'alle Charts löschen
  For Each Shp In MyDoc.Shapes
    Shp.Delete
  Next
'Neue Beschriftung
  Range("A1") = "Minimale Leitungslänge [m]"
  Range("A2") = "Maximale Leitungslänge [m]"
  Range("A3") = "Leitungsdurchmesser [mm]"
  Range("A4") = "Spez. Widerstand [Ohm mm" + Chr(178) + "/m]"
  Range("A5") = "Elektr. Spannung [V]"
  Range("A6") = "Elektr. Strom [A]"
'Ausgabetablelle
  Range("D1") = "Leitungs-"
  Range("F1") = "Spannungs-"
  Range("G1") = "Leistungs"
  Range("D2") = "Länge"
  Range("E2") = "Widerst."
  Range("F2") = "Verlust"
  Range("G2") = "Verlust"
  Range("D3") = "[m]"
  Range("E3") = "[Ohm]"
  Range("F3") = "[V]"
  Range("G3") = "[W]"
  Range("H3") = "[%]"
  Columns("B:H").Select
  Selection.NumberFormat = "0.000"
  Columns("A:A").EntireColumn.AutoFit
  Range("B1").Select
End Sub

```

End Sub

```

Sub Gleichstrom_Testdaten()

```

```

  Cells(1, 2) = 100
  Cells(2, 2) = 600
  Cells(3, 2) = 6
  Cells(4, 2) = 0.02
  Cells(5, 2) = 1000
  Cells(6, 2) = 8

```

End Sub

```

Sub Gleichstrom_Auswertung()

```

```

  Dim L1, L2, L, dL, w, R, d, A, U, I, Uv, Pv, Pp As Double
  Dim n As Integer
  L1 = Cells(1, 2)
  L2 = Cells(2, 2)
  d = Cells(3, 2)
  w = Cells(4, 2)
  U = Cells(5, 2)
  I = Cells(6, 2)
  A = d ^ 2 * 3.1415926 / 4
  dL = (L2 - L1) / 10
  n = 3
  For L = L1 To L2 Step dL
    R = w * L / A
    Uv = 2 * I * R
    Pv = 2 * I ^ 2 * R
    Pp = Pv / (U * I) * 100
    n = n + 1
    Cells(n, 4) = L
    Cells(n, 5) = R
    Cells(n, 6) = Uv
    Cells(n, 7) = Pv
    Cells(n, 8) = Pp
  Next L

```

End Sub

```

Sub Verluste_zeigen()

```

```

  Call Stromverlust_zeigen
  Call Leistungsverlust_zeigen

```

End Sub

```
Sub Stromverlust_zeigen()  
Range("F4:F14,D4:D14").Select  
Range("D4").Activate  
Charts.Add  
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers  
ActiveChart.SetSourceDataSource:= _  
    Sheets("Gleichstrom").Range("D4:D14,F4:F14"), PlotBy:=xlColumns  
ActiveChart.Location Where:=xlLocationAsObject, Name:="Gleichstrom"  
With ActiveChart  
    .HasTitle = True  
    .ChartTitle.Characters.Text = "Spannungsverluste"  
    .Axes(xlCategory, xlPrimary).HasTitle = True  
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "L [m]"  
    .Axes(xlValue, xlPrimary).HasTitle = True  
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "U [V]"  
End With  
ActiveChart.Legend.Select  
Selection.Delete  
End Sub
```

```
Sub Leistungsverlust_zeigen()  
Range("G4:G14,D4:D14").Select  
Range("D4").Activate  
Charts.Add  
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers  
ActiveChart.SetSourceData Source:=Sheets("Gleichstrom").Range("D4:D14,G4:G14" _  
    ), PlotBy:=xlColumns  
ActiveChart.Location Where:=xlLocationAsObject, Name:="Gleichstrom"  
With ActiveChart  
    .HasTitle = True  
    .ChartTitle.Characters.Text = "Leistungsverluste"  
    .Axes(xlCategory, xlPrimary).HasTitle = True  
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "L [m]"  
    .Axes(xlValue, xlPrimary).HasTitle = True  
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "P [W]"  
End With  
ActiveChart.Legend.Select  
Selection.Delete  
End Sub
```

```
Sub Rechner_Komplexe_Zahlen()  
  Load frmComCal  
  frmComCal.Show  
End Sub
```

```
Option Explicit
Private Sub cmdEnde_Click()
    Unload Me
End Sub
```

```
Private Sub cmdDelete_Click()
    TextBox1 = ""
    TextBox2 = ""
    TextBox3 = ""
    TextBox4 = ""
    TextBox5 = ""
    TextBox6 = ""
End Sub
```

```
Private Sub cmdDiv_Click()
    Dim a, a1, a2, b, b1, b2 As Double
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 + a2
    b = b1 + b2
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub
```

```
Private Sub cmdMal_Click()
    Dim a, a1, a2, b, b1, b2 As Double
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 * a2 - b1 * b2
    b = a1 * b2 + a2 * b1
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub
```

```
Private Sub cmdMinus_Click()
    Dim a, a1, a2, b, b1, b2 As Double
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 - a2
    b = b1 - b2
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub
```

```
Private Sub cmdPlus_Click()
    Dim a, a1, a2, b, b1, b2 As Double
    a1 = Val(TextBox1)
    b1 = Val(TextBox2)
    a2 = Val(TextBox3)
    b2 = Val(TextBox4)
    a = a1 + a2
    b = b1 + b2
    TextBox5 = Str(a)
    TextBox6 = Str(b)
End Sub
```

```

Option Explicit
Sub Gleichstrom_Formular()
    Dim MyDoc As Object
    Dim Shp As Shape
    Set MyDoc = ThisWorkbook.Worksheets("Wechselstrom")
    MyDoc.Activate
    MyDoc.Cells.Clear
,
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
'Neue Beschriftung
    Range("A1") = "Ohmscher Widerstand R [Ohm]"
    Range("A2") = "Kapazitiver Widerstand C [F]"
    Range("A3") = "Induktiver Widerstand L [H]"
'Ausgabetabelle
    Range("D1") = "w [1/s]"
    Range("E1") = "|Z|"
    Columns("B:H").Select
    Selection.NumberFormat = "0.000"
    Columns("A:A").EntireColumn.AutoFit
    Range("B1").Select
End Sub

Sub Wechselstrom_Testdaten()
    Cells(1, 2) = 0.8
    Cells(2, 2) = 0.002
    Cells(3, 2) = 0.01
End Sub

Sub Wechselstrom_Auswertung()
    Dim R, C, L, w, Z1, Z2, Z3, Z As Double
    Dim n As Integer
    R = Cells(1, 2)
    C = Cells(2, 2)
    L = Cells(3, 2)
    n = 1
    For w = 0 To 1000 Step 10
        Z1 = (w ^ 4 * C ^ 2 * L ^ 2 / R)
        Z2 = (w * L / R ^ 2 - w ^ 3 * C * (L / R) ^ 2 + w ^ 3 * C ^ 2 * L)
        Z3 = (w ^ 2 * C ^ 2 + (1 / R - w ^ 2 * C * L / R) ^ 2)
        Z = Sqr((Z1 ^ 2 + Z2 ^ 2) / Z3 ^ 2)
        n = n + 1
        Cells(n, 4) = w
        Cells(n, 5) = Z
    Next w
End Sub

Sub Wechselstrom_Verlauf_zeigen()
    Range("D2:E102").Select
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
    ActiveChart.SetSourceDataSource:= _
        Sheets("Wechselstrom").Range("D2:E102"), _
        PlotBy:=xlColumns
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Wechselstrom"
    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Gesamtwiderstand"
        .Axes(xlCategory, xlPrimary).HasTitle = True
        .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = _
            "Wechselspannung [1/s]"
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = _
            "Gesamtwiderstand"
    End With
    ActiveChart.Legend.Select
    Selection.Delete
End Sub

```

Option Explicit

```

Sub Regler_Formular()
    Dim MyDoc As Object
    Dim Shp As Shape
    Set MyDoc = ThisWorkbook.Worksheets("Regler")
    MyDoc.Activate
    MyDoc.Cells.Clear
,
'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
'Neue Beschriftung
    Range("A1") = "Übertragungsbeiwert KP"
    Range("A2") = "Nachstellzeit TN [s]"
    Range("A3") = "Vorhaltezeit TV [s]"
    Range("A4") = "Schrittweite " + ChrW(8710) + " t [s]"
'Ausgabetablelle
    Range("D1") = "t"
    Range("E1") = "e"
    Range("F1") = "u-P"
    Range("G1") = "u-I"
    Range("H1") = "u-D"
    Range("I1") = "u-PI"
    Range("J1") = "u-PD"
    Range("K1") = "u-PID"
    Columns("B:K").Select
    Selection.NumberFormat = "0.000"
    Columns("A:A").EntireColumn.AutoFit
    Range("B1").Select
End Sub

```

```

Sub Regler_Testdaten()
    Dim i, j As Integer
    Cells(1, 2) = 0.2
    Cells(2, 2) = 20
    Cells(3, 2) = 15
    Cells(4, 2) = 1
    For i = 1 To 100
        Cells(i + 1, 4) = i
    Next i
    j = 1
    For i = 1 To 20
        j = j + 1
        Cells(j, 5) = -10
    Next i
    For i = 1 To 20
        j = j + 1
        Cells(j, 5) = -(10 + i * 0.2)
    Next i
    For i = 1 To 41
        j = j + 1
        Cells(j, 5) = 10 - (i - 1) * 0.5
    Next i
    For i = 1 To 19
        j = j + 1
        Cells(j, 5) = 0
    Next i
End Sub

```

```

Sub Regler_Auswertung()
    Dim Kp, e, TN, TV, dt As Double
    Dim xx, t, up, e1, e2, de, u1, u2, ud, Ki As Double
    Dim i As Integer
    Kp = Cells(1, 2)
    TN = Cells(2, 2)
    TV = Cells(3, 2)
    dt = Cells(4, 2)
    u1 = 0
    For i = 1 To 100
,
'P-Anteil
        e = Cells(i + 1, 5)
        up = -Kp * e
        Cells(i + 1, 6) = up
    Next i

```

```

If i = 1 Then
    e1 = Cells(i + 1, 5)
    e2 = Cells(i + 1, 5)
Else
    e1 = Cells(i, 5)
    e2 = Cells(i + 1, 5)
End If

'I-Anteil
u2 = u1 - (e1 + e2) / 2 * Kp / TN
Cells(i + 1, 7) = u2
u1 = u2

'D-Anteil
de = e2 - e1
ud = -Kp * TV * de / dt
Cells(i + 1, 8) = ud

'PI-Anteil
Cells(i + 1, 9) = Cells(i + 1, 5) + Cells(i + 1, 6)

'PD-Anteil
Cells(i + 1, 10) = Cells(i + 1, 5) + Cells(i + 1, 7)

'PID-Anteil
Cells(i + 1, 11) = Cells(i + 1, 5) + Cells(i + 1, 6) + Cells(i + 1, 7)
Next i

'Diagramme
Call Diagramm1
Call Diagramm2
End Sub

Sub Diagramm1()
Range("D2:H101").Select
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:= _
    Sheets("Regler").Range("D2:H101"), PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = "" "Regelabweichung e" ""
ActiveChart.SeriesCollection(2).Name = "" "P-Anteil" ""
ActiveChart.SeriesCollection(3).Name = "" "I-Anteil" ""
ActiveChart.SeriesCollection(4).Name = "" "D-Anteil" ""
ActiveChart.Location Where:=xlLocationAsObject, Name:="Regler"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Regleranteile"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Zeit [s]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Regelanteile"
End With
ActiveWindow.Visible = False
Windows("Kapitel 11.xls").Activate
End Sub

Sub Diagramm2()
ActiveWindow.Visible = False
Windows("Kapitel 11.xls").Activate
Range("D2:E101,I2:K101").Select
Range("I2").Activate
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:= _
    Sheets("Regler").Range("D2:E101,I2:K101"), PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = "" "Regelabweichung e" ""
ActiveChart.SeriesCollection(2).Name = "" "PI-anteil" ""
ActiveChart.SeriesCollection(3).Name = "" "PD-Anteil" ""
ActiveChart.SeriesCollection(4).Name = "" "PID-Anteil" ""
ActiveChart.Location Where:=xlLocationAsObject, Name:="Regler"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Regler"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Zeit [s]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Regelanteil"

```

```
End With  
ActiveWindow.Visible = False  
End Sub
```

Option Explicit
Global wT(4), aT(4) As Double

Option Explicit

Sub Auswertung()

```

Dim T, M(5, 5), v(4) As Double
Dim i, il, i2, z As Integer
Dim u, u1, u2 As Double
Dim MyDoc As Object
Dim Shp As Shape
Set MyDoc = ThisWorkbook.Worksheets("Fuzzy")
MyDoc.Activate
MyDoc.Cells.Clear

```

'alle Charts löschen

```

For Each Shp In MyDoc.Shapes
    Shp.Delete
Next

```

'Auswertungsstart

z = 0

'Über alle Temperaturen

For T = -10 To 50

'Fuzzyifizierung

Call FuzzyTemperatur(T)

For i = 0 To 4

M(i + 1, 0) = wT(i)

Next i

z = z + 1

Cells(z, 1) = T

'Über alle Temperaturänderungen

For i = 1 To 17

Call FuzzyÄnderung(i)

For il = 0 To 4

M(0, il + 1) = aT(il)

Next il

'Inferenzen

For il = 1 To 5

For i2 = 1 To 5

M(il, i2) = Minimum(M(il, 0), M(0, i2))

Next i2

Next il

'Defuzzyifizierung

$$v(4) = M(1, 1) + M(1, 2) + M(1, 3) + _ \\ M(2, 1) + M(2, 2) + M(3, 1)$$

$$v(3) = M(1, 4) + M(2, 3) + M(3, 2) + M(4, 1)$$

$$v(2) = M(1, 5) + M(2, 4) + M(3, 3) + _ \\ M(4, 2) + M(5, 1)$$

$$v(1) = M(2, 5) + M(3, 4) + M(4, 3) + M(5, 2)$$

$$v(0) = M(3, 5) + M(4, 4) + M(4, 5) + _ \\ M(5, 3) + M(5, 4) + M(5, 5)$$

sv = v(0) + v(1) + v(2) + v(3) + v(4)

For il = 0 To 4

v(il) = v(il) / sv

Next il

$$u1 = v(1) ^ 2 * 0.25 + v(2) ^ 2 * 0.5 + v(3) ^ 2 * 0.75 + v(4) ^ 2 * 1$$

$$u2 = v(0) ^ 2 + v(1) ^ 2 + v(2) ^ 2 + v(3) ^ 2 + v(4) ^ 2$$

u = u1 / u2

Cells(z, 1 + i) = u

Next i

Next T

Call FuzzyKennfeld

End Sub

Function Minimum(a, b) As Double

If a <= b Then

Minimum = a

Else

Minimum = b

End If

End Function

Sub FuzzyTemperatur(T)

Dim i As Integer

```

If T < 0 Then
    wT(0) = 1
Else
    If T <= 10 Then
        wT(0) = (10 - T) / 10
    Else
        wT(0) = 0
    End If
End If
For i = 1 To 3
    If T < (i - 1) * 10 Then
        wT(i) = 0
    Else
        If T <= i * 10 Then
            wT(i) = (T - (i - 1) * 10) / 10
        Else
            If T <= (i + 1) * 10 Then
                wT(i) = ((i + 1) * 10 - T) / 10
            Else
                wT(i) = 0
            End If
        End If
    End If
End If
Next i
If T < 30 Then
    wT(4) = 0
Else
    If T <= 40 Then
        wT(4) = (T - 30) / 10
    Else
        wT(4) = 1
    End If
End If
End Sub

Sub FuzzyÄnderung(i)
    Select Case i
    Case 1
        aT(0) = 1: aT(1) = 0: aT(2) = 0: aT(3) = 0: aT(4) = 0
    Case 2
        aT(0) = 0.75: aT(1) = 0.25: aT(2) = 0: aT(3) = 0: aT(4) = 0
    Case 3
        aT(0) = 0.5: aT(1) = 0.5: aT(2) = 0: aT(3) = 0: aT(4) = 0
    Case 4
        aT(0) = 0.25: aT(1) = 0.75: aT(2) = 0: aT(3) = 0: aT(4) = 0
    Case 5
        aT(0) = 0: aT(1) = 1: aT(2) = 0: aT(3) = 0: aT(4) = 0
    Case 6
        aT(0) = 0: aT(1) = 0.75: aT(2) = 0.25: aT(3) = 0: aT(4) = 0
    Case 7
        aT(0) = 0: aT(1) = 0.5: aT(2) = 0.5: aT(3) = 0: aT(4) = 0
    Case 8
        aT(0) = 0: aT(1) = 0.25: aT(2) = 0.75: aT(3) = 0: aT(4) = 0
    Case 9
        aT(0) = 0: aT(1) = 0: aT(2) = 1: aT(3) = 0: aT(4) = 0
    Case 10
        aT(0) = 0: aT(1) = 0: aT(2) = 0.75: aT(3) = 0.25: aT(4) = 0
    Case 11
        aT(0) = 0: aT(1) = 0: aT(2) = 0.5: aT(3) = 0.5: aT(4) = 0
    Case 12
        aT(0) = 0: aT(1) = 0: aT(2) = 0.25: aT(3) = 0.75: aT(4) = 0
    Case 13
        aT(0) = 0: aT(1) = 0: aT(2) = 0: aT(3) = 1: aT(4) = 0
    Case 14
        aT(0) = 0: aT(1) = 0: aT(2) = 0: aT(3) = 0.75: aT(4) = 0.25
    Case 15
        aT(0) = 0: aT(1) = 0: aT(2) = 0: aT(3) = 0.5: aT(4) = 0.5
    Case 16
        aT(0) = 0: aT(1) = 0: aT(2) = 0: aT(3) = 0.25: aT(4) = 0.75
    Case 17
        aT(0) = 0: aT(1) = 0: aT(2) = 0: aT(3) = 0: aT(4) = 1
    End Select
End Sub

Sub FuzzyKennfeld()
    Range("B1:R61").Select
    Charts.Add

```

```

ActiveChart.ChartType = xlSurfaceWireframe
ActiveChart.SetSourceData Source:=Sheets("Fuzzy").Range("B1:R61"), _
    PlotBy:= xlColumns
ActiveChart.SeriesCollection(1).Name = ""sinkt schnell""
ActiveChart.SeriesCollection(5).Name = ""sinkt""
ActiveChart.SeriesCollection(9).Name = ""stabil""
ActiveChart.SeriesCollection(13).Name = ""steigt""
ActiveChart.SeriesCollection(17).Name = ""steigt schnell""
ActiveChart.Location Where:=xlLocationAsObject, Name:="Fuzzy"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Fuzzy Regelfeld"
    .Axes(xlCategory).HasTitle = True
    .Axes(xlCategory).AxisTitle.Characters.Text = "Temperatur [grdC]"
    .Axes(xlSeries).HasTitle = True
    .Axes(xlSeries).AxisTitle.Characters.Text = "Änderung"
    .Axes(xlValue).HasTitle = True
    .Axes(xlValue).AxisTitle.Characters.Text = "Ventilstellung"
End With
ActiveChart.HasLegend = False
ActiveSheet.Shapes(1).ScaleWidth 1.4, msoFalse, _
    msoScaleFromBottomRight
ActiveSheet.Shapes(1).ScaleHeight 1.44, msoFalse, _
    msoScaleFromBottomRight
ActiveSheet.Shapes(1).ScaleWidth 1.24, msoFalse, _
    msoScaleFromTopLeft
ActiveSheet.Shapes(1).ScaleHeight 1.22, msoFalse, _
    msoScaleFromTopLeft
ActiveChart.Walls.Select
With Selection.Border
    .ColorIndex = 16
    .Weight = xlThin
    .LineStyle = xlContinuous
End With
With Selection.Interior
    .ColorIndex = 2
    .PatternColorIndex = 1
    .Pattern = xlSolid
End With
ActiveChart.Floor.Select
With Selection.Border
    .Weight = xlHairline
    .LineStyle = xlAutomatic
End With
With Selection.Interior
    .ColorIndex = 2
    .PatternColorIndex = 2
    .Pattern = xlSolid
End With
ActiveChart.Walls.Select
ActiveChart.Corners.Select
With ActiveChart
    .Elevation = 25
    .Rotation = 150
End With
With ActiveChart
    .Elevation = 29
    .Rotation = 155
End With
ActiveChart.Axes(xlSeries).AxisTitle.Select
Selection.Left = 22
Selection.Top = 357
ActiveChart.Deselect
ActiveChart.Corners.Select
With ActiveChart
    .Elevation = 29
    .Rotation = 154
End With
With ActiveChart
    .Elevation = 30
    .Rotation = 152
End With
ActiveChart.ChartArea.Select
ActiveChart.Axes(xlCategory).Select
ActiveChart.Corners.Select
With ActiveChart
    .Elevation = 32

```

```
.Rotation = 153
End With
With ActiveChart
    .Elevation = 38
    .Rotation = 160
End With
ActiveChart.ChartArea.Select
ActiveWindow.Visible = False
Windows("Kapitel 11.xls").Activate
Range("Q51").Select
End Sub
```